ISO TC 184/SC4/WG3_ N728___ **Date:** 1998-01-30

Supersedes ISO TC 184/SC4/WG3 N726____

ISO/WD 10303-223

Product data reprsentation and exchange: Application protocol: Exchange of design and manufacturing product information for cast parts

ABSTRACT:

KEYWORDS:

COMMENTS TO READER:

This is version 0.18 of the AP.

Project Leader: Gerald M. Radack **Address:** Concurrent Technologies Corp. 1450 Scalp Avenue Johnstown, PA 15904 USA **Telephone:** +1 814 269 2679 **Telefacsimile:** +1 814 269 2402 **Electronic mail:** radack@ctc.com	**Project Editor:** Somendra Singh **Address:** Concurrent Technologies Corp. 1450 Scalp Avenue Johnstown, PA 15904 USA **Telephone:** +1 814 269 2434 **Telefacsimile:** +1 814 269 2402 **Electronic mail:** singh@ctc.com

Contents Page

Annexes

Figures

Tables

Foreword

The International Organization for Standardization (ISO) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

International Standard ISO 10303–223 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration,* Subcommittee SC4, *Industrial data.*

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration – Product data representation and exchange:*

- Part 1, Overview and fundamental principles;
- Part 11, Description method: EXPRESS language reference manual;
- Part 12, Description method: EXPRESS-I language reference manual;
- Part 13, Description method: Architecture and methodology reference manual;
- Part 21, Implementation method: Clear text encoding of the exchange structure;
- Part 22, Implementation method: Standard data access interface specification;
- Part 23, Implementation method: C++ language binding to the standard data access interface;
- Part 24, Implementation method: C language binding to the standard data access interface;
- Part 26, Implementation method: Interface definition language binding to the standard data access interface;
- Part 31, Conformance testing methodology and framework: General concepts;
- Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;
- Part 33, Conformance testing methodology and framework: Structure and use of abstract test suites;
- Part 34, Conformance testing methodology and framework: Abstract test methods;

- Part 35, Conformance testing methodology and framework: Abstract test methods for SDAI implementations;
- Part 41, Integrated generic resource: Fundamentals of product description and support;
- Part 42, Integrated generic resource: Geometric and topological representation;
- Part 43, Integrated generic resource: Representation structures;
- Part 44, Integrated generic resource: Product structure configuration;
- Part 45, Integrated generic resource: Materials;
- Part 46, Integrated generic resource: Visual presentation;
- Part 47, Integrated generic resource: Shape variation tolerances;
- Part 49, Integrated generic resource: Process structure and properties;
- Part 101, Integrated application resource: Draughting;
- Part 104, Integrated application resource: Finite element analysis;
- Part 105, Integrated application resource: Kinematics;
- Part 106, Integrated application resource: Building construction core model;
- Part 201, Application protocol: Explicit draughting;
- Part 202, Application protocol: Associative draughting;
- Part 203, Application protocol: Configuration controlled design;
- Part 204, Application protocol: Mechanical design using boundary representation;
- Part 205, Application protocol: Mechanical design using surface representation;
- Part 207, Application protocol: Sheet metal die planning and design;
- Part 208, Application protocol: Life cycle management - Change process;
- Part 209, Application protocol: Composite and metallic structural analysis and related design;
- Part 210, Application protocol: Design of layered electronic products;
- Part 211, Application protocol: Electronics test diagnostics and remanufacture;

- Part 212, Application protocol: Electrotechnical design and installation;
- Part 213, Application protocol: Numerical control process plans for machined parts;
- Part 214, Application protocol: Core data for automotive mechanical design;
- Part 215, Application protocol: Ship arrangement;
- Part 216, Application protocol: Ship moulded forms;
- Part 217, Application protocol: Ship piping;
- Part 218, Application protocol: Ship structures;
- Part 220, Application protocol: Process planning, manufacture, and assembly of layered electronic products;
- Part 221, Application protocol: Functional data and their schematic representation for process plant;
- Part 222, Application protocol: Exchange of product data for composite structures;
- Part 223, Application protocol: Exchange of design and manufacturing product information for cast parts;
- Part 224, Application protocol: Mechanical product definition for process plans using mechanical feature;
- Part 225, Application protocol: Building elements using explicit shape representation;
- Part 226, Application protocol: Ship mechanical systems;
- Part 227, Application protocol: Plant spatial configuration;
- Part 228, Application protocol: Building services: Heating, ventilation, and air conditioning;
- Part 229, Application protocol: Exchange of design and manufacturing product information for forged parts;
- Part 230, Application protocol: Building structural frame: Steelwork;
- Part 231, Application protocol: Process engineering data: Process design and process specification of major equipment;
- Part 232, Application Protocol: Technical data package;

- Part 301, Abstract test suite: Explicit draughting;
- Part 302, Abstract test suite: Associative draughting;
- Part 303, Abstract test suite: Configuration controlled design;
- Part 304, Abstract test suite: Mechanical design using boundary representation;
- Part 305, Abstract test suite: Mechanical design using surface representation;
- Part 307, Abstract test suite: Sheet metal die planning and design;
- Part 308, Abstract test suite: Life cycle management - Change process;
- Part 309, Abstract test suite: Composite and metallic structural analysis and related design;
- Part 310, Abstract test suite: Design of layered electronic products;
- Part 311, Abstract test suite: Electronics test diagnostics and remanufacture;
- Part 312, Abstract test suite: Electrotechnical design and installation;
- Part 313, Abstract test suite: Numerical control process plans for machined parts;
- Part 314, Abstract test suite: Core data for automotive mechanical design;
- Part 315, Abstract test suite: Ship arrangement;
- Part 316, Abstract test suite: Ship moulded forms;
- Part 317, Abstract test suite: Ship piping;
- Part 318, Abstract test suite: Ship structures;
- Part 320, Abstract test suite: Process planning, manufacture, and assembly of layered electronic products;
- Part 321, Abstract test suite: Functional data and their schematic representation for process plant;
- Part 322, Abstract test suite: Exchange of product data for composite structures;
- Part 323, Abstract test suite: Exchange of design and manufacturing product information for cast parts;
- Part 324, Abstract test suite: Mechanical product definition for process plans using mechanical features;

– Part 325, Abstract test suite: Building elements using explicit shape representation;
– Part 326, Abstract test suite: Ship mechanical systems;
– Part 327, Abstract test suite: Plant spatial configuration;
– Part 328, Abstract test suite: Building services: Heating, ventilation, and air conditioning;
– Part 329, Abstract test suite: Exchange of design and manufacturing product information for forged parts;
– Part 330, Abstract test suite: Building structural frame: Steelwork;
– Part 331, Abstract test suite: Process engineering data: Process design and process specification of major equipment;
– Part 332, Abstract test suite: Technical data package;
– Part 501, Application interpreted construct: Edge-based wireframe;
– Part 502, Application interpreted construct: Shell-based wireframe;
– Part 503, Application interpreted construct: Geometrically bounded 2D wireframe;
– Part 504, Application interpreted construct: Draughting annotation;
– Part 505, Application interpreted construct: Drawing structure and administration;
– Part 506, Application interpreted construct: Draughting elements;
– Part 507, Application interpreted construct: Geometrically bounded surface;
– Part 508, Application interpreted construct: Non-manifold surface;
– Part 509, Application interpreted construct: Manifold surface;
– Part 510, Application interpreted construct: Geometrically bounded wireframe;
– Part 511, Application interpreted construct: Topologically bounded surface;
– Part 512, Application interpreted construct: Faceted boundary representation;
– Part 513, Application interpreted construct: Elementary boundary representation;
– Part 514, Application interpreted construct: Advanced boundary representation;
– Part 515, Application interpreted construct: Constructive solid geometry;

- Part 517, Application interpreted construct: Mechanical design geometric presentation;
- Part 518, Application interpreted construct: Mechanical design shaded representation.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of this International Standard reflects its structure:

- Parts 11 to 13 specify the description methods,
- Parts 21 to 26 specify the implementation methods,
- Parts 31 to 35 specify the conformance testing methodology and framework,
- Parts 41 to 49 specify the integrated generic resources,
- Parts 101 to 106 specify the integrated application resources,
- Parts 201 to 232 specify the application protocols,
- Parts 301 to 332 specify the abstract test suites, and
- Parts 501 to 518 specify the application interpreted constructs.

Should further parts of ISO 10303 be published, they will follow the same numbering pattern.

Annexes A,B,C,D and E form an integral part of this part of ISO 10303. Annexes F,G, H, and J are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the application protocol series.

This part of ISO 10303 specifies an application protocol (AP) for the use of product data within a defined context which satisfies an industrial need to exchange product and process design and manufacturing data of cast parts.

This application protocol defines the context, scope, and information requirements for the exchange of design and manufacturing product information for cast parts and specifies the integrated resources necessary to satisfy these requirements.

Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for the conformance testing of AP implementations.

Clause 1 defines the scope of the application protocol and summarizes the functionality and data covered by the AP. An application activity model that is the basis for the definition of the scope is provided in annex F. The information requirements for the application are specified in clause 4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in annex G.

Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in clause 5.1 shows the correspondence between the information requirements and the AIM. The short listing of the AIM specifies the interface to the integrated resources and is given in 5.2. Note that the definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. The expanded listing given in Annex A contains the complete EXPRESS for the AIM without annotation. A graphical representation of the AIM is given in Annex H. Additional requirements for specific implementation methods are given in annex C.

Design and manufacturing product information for casting parts describes the product and process information required to produce a casting which meets the design requirements. Information concerning manufacture by a casting process is necessary for the complete data exchange between systems used by product designers, foundry men, and pattern makers. In addition, material information for the application of process expertise is supported. The AP describes the design resulting from the casting expert's decision process rather than the decision process itself. Product definition includes solid geometry and topology, configuration information, material specification, tolerance requirements, casting form features, surface finish information, and manufacturing process data. Also included in this AP is the process data associated with manufacturing a casting part: casting with rigging, mold information, pour parameters, and inspection results. The AP includes modifications to the part shape and the items connected as necessary to complete the cavity for filling and molding the casting product. Figure 1 contains the data planning model that provides a high-level description of the requirements for the application protocol, as well as the relationship between the information.

Industrial automation systems and integration — Product data representation and exchange — Part 223 : Application Protocol: Exchange of design and manufacturing product information for cast parts

1 Scope

This part of ISO 10303 specifies the exchange, archiving and sharing of design and manufacturing product information for cast parts.

NOTE – The application activity model in annex E provides a graphical representation of the processes and information flows which are the basis for the definition of the scope of this part of ISO 10303.

The following are within the scope of this part of ISO 10303:

– Parts and process plans for parts that are made by sand, die, and investment casting processes;

– Design data for cast parts, including geometry, materials, tolerances, required physical and mechanical properties, required tests;

– Characterization of products used to make cast parts, including molds, dies, equipment, materials, and consumable items;

– Specifications for patterns and die assemblies;

– Input to and output from casting process simulation software;

– Data exchange between customer and foundry, within the foundry, and between foundry and supplier;

– Use of data for foundry automation and shop floor control;

– Use of data for archiving of design and manufacturing data for cast parts.

NOTE – Data supported by this AP may need to be archived to meet legal and regulatory requirements, and to meet quality objectives.

The following are outside the scope of this part of ISO 10303:

– Data describing rules, guidelines and expert knowledge used to design and manufacture cast parts;

– Data describing why a particular design or manufacturing decision was made;

- Shop floor scheduling data;
- Process plans for making patterns, dies, and other tooling;
- Algorithms used to obtain simulation results.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 31:1994, *Quantities and Units*

ISO 1000:1992, *SI units and recommendations for the use of their multiples and of certain other units*

ISO/IEC 8824–1:Not yet published, *Information Technology - Open Systems Interconnection - Abstract Syntax Notation One (ASN.1) - Part 1:Specification of Basic Notation.*

ISO 10303–1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles.*

ISO 10303–11: 1994, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.*

ISO 10303–21: 1994, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure.*

ISO 10303–31:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 31: Conformance testing methodology and framework: General concepts.*

ISO 10303–41:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resources: Fundamentals of product description and support.*

ISO 10303–42:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resources: Geometric and topological representation.*

ISO 10303–43:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resources: Representation structures.*

ISO 10303–44:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 44: Integrated generic resources: Product structure configuration.*

ISO 10303–45:Not yet published, *Industrial automation systems and integration — Product data representation and exchange — Part 45: Integrated generic resources: Materials.*

ISO 10303–46:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 46: Integrated generic resources: Visual presentation.*

ISO 10303–47:1997, *Industrial automation systems and integration — Product data representation and exchange — Part 47: Integrated generic resources: Shape variation tolerances.*

ISO 10303–49:Not yet published, *Industrial automation systems and integration — Product data representation and exchange — Part 49: Integrated generic resources: Process stucture and properties.*

ISO 10303–104:Not yet published, *Industrial automation systems and integration — Product data representation and exchange — Part 104: Integrated application resources: Finite element analysis.*

3 Definitions and abbreviations

For the purposes of this part of ISO 10303, the following definitions and abbreviations apply.

3.1 Terms defined in ISO 8062-1

This Part of ISO 10303 makes use of the following terms defined in ISO 8062-1.

- draft angle;
- parting surface;
- taper.

3.2 Terms defined in ISO 10303-1

This Part of ISO 10303 makes use of the following terms defined in ISO 10303-1.

- abstract test suite;
- application;

- application activity model;
- application context;
- application interpreted model;
- application object;
- application protocol;
- application reference model;
- assembly;
- conformance testing;
- context;
- data;
- data exchange;
- implementation method;
- information;
- integrated resource;
- interpretation;
- model;
- product;
- product data;
- structure;
- unit of functionality.

3.3 Terms defined in ISO 10303-31

This Part of ISO 10303 makes use of the following terms defined in ISO 10303-31.

- conformance testing;
- preprocessor;

– postprocessor.

3.4 Terms defined in ISO 10303-42

This Part of ISO 10303 makes use of the following terms defined in ISO 10303-42.

- arc;
- arcwise connected;
- axi-symmetric;
- bounds;
- boundary;
- boundary representation solid model;
- closed curve;
- closed surface;
- connected;
- connected component;
- curve;
- cycle;
- dimensionality;
- domain;
- extent;
- finite;
- geometric coordinate system;
- graph;
- handle;
- homomorphic;
- inside;

- list;
- d-manifold with boundary;
- open curve;
- open surface;
- orientable;
- overlap;
- parameter range;
- parameter space;
- placement coordinate system;
- self-intersect;
- self-loop;
- set;
- space dimensionality;
- surface;
- topological sense.

3.5 Other definitions

For the purpose of this Part of ISO 10303, the following definitions apply.

3.5.1 die casting: a category of casting processes which use permanent, steel or iron, molds. In this type of casting process one die makes numerous castings.

> NOTE – This category includes high and low pressure die casting processes. The high pressure processes include normal die casting, squeeze casting, semi-solid casting and vacuum die casting; and usually use large, complex molds constructed of heat treated tool steel or other high strength materials. The low pressure processes use a variety of methods to introduce metal into the mold, including gravity pour, vacuum and pressure lift, and tilt pour processes. Because high pressures are not employed, special cast iron alloys are usually used to construct the molds.

3.5.2 investment casting: a category of casting processes which coat or invest a consumable wax pattern with the refractory material.

NOTE – This category includes the lost wax investment process, where a consumable wax pattern is coated or invested with a plaster or ceramic shell mold. Also, plastics and foams have been used to produce the pattern. This latter process is also included in this application protocol and is called lost foam, expendable pattern or evaporative pattern casting.

3.5.3 sand casting: a category of casting processes which use a pattern, also called a match plate or pressure plate, to make a disposable mold for each casting poured.

NOTE – This category includes and covers all unbonded green sand processes, bonded sand processes, and plaster mold processes. The term "sand" used in this application protocol includes green sand, bonded sand, plaster, and other similar molding materials used to create a mold. This category also includes semi-permanent molding processes, in the sense that the permanent part of the mold is treated as a specially designed chill. Also included is the sand core package assembly casting process, where a number of bonded sand cores are assembled to form a continuous cavity. In this case no pattern is employed. The pattern in effect divided up among the different cores.

3.5.4 boundary representation model: a solid model represented by the set of shells defining their exterior or interior boundaries. Constraints ensure that the associated geometry is well defined and that the Euler formula connecting the numbers of vertices, edges, faces, loops and shells in the model is satisfied.

3.5.5 component: a product identified from a group of associated products.

3.5.6 design phase: the period during which the engineering representation of a product is dynamic.

3.5.7 mechanical part: a physical object which can be formed from material into a static shape.

3.5.8 solid model: a three dimensional object with a well defined inside and outside separated by a two dimensional boundary.

3.6 Abbreviations

For the purposes of this Part of ISO 10303, the following abbreviations apply.

AAM	application activity model
AIM	application interpreted model
AP	application protocol
ARM	application reference model

4 Information requirements

This clause specifies the information required for the exchange of design and manufacturing product information for cast parts.

The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using the terminology of the subject area of this application protocol.

NOTES

1 – A graphical representation of the information requirements is given in annex G.

2 – The information requirements correspond to those of the activities identified as being in the scope of this application protocol in annex F.

3 – The mapping table specified in 5.1 shows how the information requirements are met using the integrated resources of this International Standard. The use of the integrated resources introduces additional requirements that are common to application protocols.

4.1 Units of functionality

This subclause specifies the units of functionality for the Exchange of design and manufacturing product information for cast parts application protocol. This Part of ISO 10303 specifies the following units of functionality:

- basic_data;
- configuration_management;
- die_casting;
- feature;
- geometry;
- general_item;
- investment_casting;
- measurement;
- process_plan;
- property;
- quality_assurance;
- quality_control;

- requirement;
- sand_casting;
- simulation;
- substance_composition;
- tolerance.

The units of functionality and a description of the functions that each UoF supports are given below. The application objects included in the UoFs are defined in **??**

4.1.1 basic_data

The basic_data UoF specifies the data constructs to organize the design and manufacturing information.

The following application objects are used by the basic_data UoF:

- Computer_file;
- Curve_point;
- Data_curve;
- Date_and_time;
- Document;
- Graphics;
- Organization;
- Person_and_organization;
- Picture;
- Specification.

4.1.2 configuration_management

The configuration_management UoF specifies the configuration management and control of the items and process plans used in casting.

The following application objects are used by the configuration_management UoF:

- Approval;

- Date;
- Design_request;
- Design_update;
- Item;
- Item_version;
- Process_plan_version;
- Request_for_change;
- Request_for_clarification;
- Version.

4.1.3 die_casting

The die_casting UoF specifies the items required for the design and manufacture of casting products using die casting technique.

The following application objects are used by the die_casting UoF:

- Die_component;
- Die_mold_assembly;
- Die_sub_assembly.

4.1.4 feature

The feature UoF specifies the definitions and representations for the casting application domain features which describe characteristics of the casting product significant to the casting process design.

The following application objects are used by the feature UoF:

- Casting_design_feature;
- Casting_machining_allowance;
- Casting_machining_allowance_along_normal;
- Casting_machining_allowance_along_vector;
- Casting_machining_with_explicit_surface;

- Casting_round_corner_transition;
- Casting_round_edge_transition;
- Casting_taper;
- Feature_parameter;
- Identification_marking;
- Parting_surface.

4.1.5 geometry

The geometry UoF specifies the shape representation of the items used in the casting processes.

The following application objects are used by the geometry UoF:

- Annotation;
- Axis_placement;
- B_rep_shape_representation;
- Curve_on_surface;
- Geometric_2d_shape_representation;
- Shape_aspect;
- Shape_representation;
- Vector.

4.1.6 general_item

The general_item UoF specifies the definitions for basic casting features.

The following application objects are used by the general_item UoF:

- Assembly_item;
- Assembly_item_placement;
- Casting_part;
- Consumable_item;
- Core;

- Core_master;
- Customer_part;
- Finished_casting;
- In_mold_rigging_component;
- Insert;
- Master;
- Pattern_master;
- Raw_casting;
- Secondary_tooling;
- Simple_item;
- Update_relationship.

4.1.7 investment_casting

The investment_casting UoF specifies the items required for the design and manufacture of the casting products using investment casting technique.

The following application objects are used by the investment_casting UoF:

- Investment_mold;
- Investment_pattern;
- Investment_pattern_assembly;
- Investment_pattern_die;
- Investment_pattern_die_component;
- Investment_sprue_assmebly;
- Investment_sprue_component.

4.1.8 measurement

The measurement UoF specifies the data constructs of the value and unit components for representing the measurable properties.

The following application objects are used by the measurement UoF:

- Measure_range_with_unit;
- Measure_with_unit;
- Scalar_measure;
- Statistical_measure;
- Tensor_measure;
- Unit;
- Vector_measure.

4.1.9 process_plan

The process_plan UoF specifies the data constructs to describe the operations, input items, output items, and equipment usage needed for producing the cast parts.

The following application objects are used by the process_plan UoF:

- Compound_process;
- Equipment_usage;
- Machine_setting;
- Process;
- Process_plan;
- Substance_usage;
- Unit_process.

4.1.10 property

The property UoF the data constructs used to represent the property value and the property value relationship.

The following application objects are used by the property UoF:

- Property_relationship;
- Property_value.

4.1.11 quality_assurance

The quality_assurance UoF specifies the inspection and test results obtained from a sample set of lots or heats of castings.

The following application objects are used by the quality_assurance UoF:

- Artifact;
- Failure_report;
- Heat;
- Inspection_or_test_result;
- Lot;
- Sampled_set.

4.1.12 quality_control

The quality_control UoF specifies the process execution records with respect to the processes described in the process plan. The records include the actual values used to perform the processes.

The following application objects are used by the quality_control UoF:

- Machine_setting_record;
- Process_execution;
- Process_execution_record;
- Process_parameter_record;
- Substance_composition_element_record;
- Substance_usage_record.

4.1.13 requirement

The requirement UoF specifies the items required for specification, design and manufacture of the casting product.

The following application objects are used by the requirement UoF:

- Composition_requirement;
- Geometry_requirement;

- Heat_treat_requirement;
- Inspection_or_test_requirement;
- Item_requirement;
- Process_requirement;
- Property_requirement;
- Reporting_requirement;
- Required_machining_allowance;
- Specification_reference;
- Surface_roughness_requirement.

4.1.14 sand_casting

The sand_casting UoF specifies the items required for the design and manufacture of the casting product using sand casting technique.

The following application objects are used by the sand_casting UoF:

- Flask;
- Sand_mold;
- Sand_mold_assembly;
- Sand_mold_component;
- Pattern_and_rigging;
- Pattern;
- Pattern_plate;
- Pattern_rigging_component.

4.1.15 simulation

The simulation UoF specifies the input data and the output data for a casting process simulation model.

The following application objects are used by the simulation UoF:

- Boundary_condition;
- Condition_or_assumption;
- Integration_interval;
- Meshing_condition;
- Simulated_property;
- Simulation_input;
- Simulation_input_region;
- Simulation_output;
- Simulation_output_region;
- Simulation_result;
- Simulation_run;
- Simulation_run_relationship;
- Simulation_software;
- Simulation_unit;
- Simulation_unit_state.

4.1.16 substance_composition

The substance_composition UoF specifies the data constructs for representing the physical and chemical compositions of substances used in casting.

The following application objects are used by the substance_composition UoF:

- Geometry_characterization;
- Substance;
- Substance_composition_element;
- Substance_structure;
- Substance_structure_element.

4.1.17 tolerance

The tolerance UoF specifies the allowable deviation from nominal or ideal conditions. Geometric tolerance, dimensional tolerance, and mismatch tolerance for two half dies are included.

The following application objects are used by the tolerance UoF:

- Datum_reference_frame;
- Dimensional_tolerance;
- Geometric_tolerance;
- Mismatch_tolerance;
- Tolerance.

4.2 Application objects

This subclause specifies the application objects for the Exchange of design and manufacturing product information for cast parts application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

4.2.1 Annotation

An Annotation is additional human interpretable information associated with a piece of product data. The data associated with an Annotation are the following:

- applies_to;
- description;
- more_information;
- name.

4.2.1.1 applies_to

The applies_to specifies the defined portion of a geometric shape to which the Annotation pertains.

4.2.1.2 description

The description specifies the textual account of an Annotation.

4.2.1.3 more_information

The more_information specifies the document that supplements the description of an Annotation. The more_information need not be specified for a particular Annotation.

4.2.1.4 name

The name specifies the human-understandable term used to identify an Annotation. The name need not be specified for a particular Annotation.

4.2.2 Approval

An Approval is an indication of concurrence or nonconcurrence with a piece of product data. The data associated with an Approval are the following:

- approved_by;
- approved_version;
- when_approved.

4.2.2.1 approved_by

The approved_by specifies the individual who has approved a piece of product data.

4.2.2.2 approved_version

The approved_version specifies the version that has been approved.

4.2.2.3 when_approved

The when_approved specifies the Date_and_time (see 4.2.30) at which the Approval became, or will become, effective.

4.2.3 Artifact

An Artifact is a physical object that has been produced by the casting process. The data associated with an Artifact are the following:

- belongs_to_heat;
- belongs_to_lot;
- part_number;
- process_plan_used;

- product_specification_used;
- production_date.

4.2.3.1 belongs_to_heat

The belongs_to_heat specifies the Heat (see 4.2.50) to which an Artifact belongs. The belongs_to_heat need not be specified for a particular Artifact.

4.2.3.2 belongs_to_lot

The belongs_to_lot specifies the Lot (see 4.2.69) to which an Artifact belongs. The belongs_to_lot need not be specified for a particular Artifact.

4.2.3.3 part_number

The part_number specifies the label used to identify an Artifact.

4.2.3.4 process_plan_used

The process_plan_used specifies the operations by which the Artifact was manufactured.

4.2.3.5 product_specification_used

The product_specification_used specifies the design that the Artifact was built to satisfy.

4.2.3.6 production_date

The production_date specifies the Date_and_time (see 4.2.30) when the Artifact was created.

4.2.4 Assembly_item

An Assembly_item is a type of Item (see 4.2.66) that is constructed by joining together simpler items. Each Assembly_item is either a Die_mold_assembly (see 4.2.36), a Die_sub_assembly (see 4.2.37), an Investment_pattern_assembly (see 4.2.59), an Investment_pattern (see 4.2.60), an Investment_pattern_die (see 4.2.62), an Investment_sprue_assembly (see 4.2.63), a Pattern_and_rigging (see 4.2.79), a Sand_mold_assembly (see 4.2.102), or a Sand_mold (see 4.2.104). The data associated with an Assembly_item are the following:

- elements.

The elements specifies the items that are part of the Assembly_item and their placement relative to the coordinate system of the Assembly_item. There may be more than one elements for an Assembly_item.

4.2.5 Assembly_item_placement

An Assembly_item_placement is the location of a Item (see 4.2.66) with respect to the coordinate system of an Assembly_item (see 4.2.4) in which it is contained. The data associated with an Assembly_item_placement are the following:

- location;
- placed_item.

4.2.5.1 location

The location specifies the coordinate transformation that locates the placed_item with respect to its parent.

4.2.5.2 placed_item

The placed_item specifies the Item (see 4.2.66) that is to be placed at a specific location within an assembly.

4.2.6 Axis_placement

An Axis_placement is the location and orientation of a geometric item with respect to some coordinate system.

4.2.7 B_rep_shape_representation

A B_rep_shape_representation is a type of Shape_representation (see 4.2.108) in which the size and shape of the solid is defined in terms of the faces, edges and vertices which make up its boundary.

4.2.8 Boundary_condition

A Boundary_condition is the state of physical phenomena at the surface of a region.

NOTE – Boundary conditions include heat transfer coefficients and pressure on mold and vents.

The data associated with a Boundary_condition are the following:

- applies_to;
- condition_value.

4.2.8.1 applies_to

The applies_to specifies the surface to which the Boundary_condition pertains.

4.2.8.2 condition_value

The condition_value specifies the property value or relationship between property values that characterizes the Boundary_condition.

4.2.9 Casting_design_feature

A Casting_design_feature is a geometric form which conforms to some preconceived pattern or stereotype and is, for the purpose of casting design, usefully dealt with as an occurrence of that stereotype.

EXAMPLES

1 – A fillet is required when two plane segments join at a "T" section in order to prevent shrinkage porosity from occurring at the joint.

2 – A taper (or draft angle) is required on walls parallel to the opening direction of a die so that a casting may be easily pulled away from the die without sticking or tearing.

Each Casting_design_feature is either a Casting_machining_allowance (see 4.2.10), a Casting_round_corner_transition (see 4.2.15), a Casting_round_edge_transition (see 4.2.16), a Casting_taper (see 4.2.17), or an Identification_marking (see 4.2.52). The data associated with a Casting_design_feature are the following:

- geometry_to_be_modified;
- resultant_geometry.

4.2.9.1 geometry_to_be_modified

The geometry_to_be_modified specifies the shape of a part without the feature.

4.2.9.2 resultant_geometry

The resultant_geometry specifies the shape of a part after the feature has been applied.

4.2.10 Casting_machining_allowance

A Casting_machining_allowance is a type of Casting_design_feature (see 4.2.9) that is a machining allowance as defined in ISO 8062-1.

NOTES

1 – ISO 8062-1 defines machining allowance to be "allowance left on a raw moulded part so that material and surface imperfections can be removed by subsequent machining to achieve the specified surface texture and linear and geometrical dimensions."

2 – A cast part is a kind of moulded part.

Each Casting_machining_allowance may be one of the following: a Casting_machining_allowance_along_normal (see 4.2.11), a Casting_machining_allowance_along_vector (see 4.2.12), or a Casting_machining_allowance_with_explicit_surface (see 4.2.13).

4.2.11 Casting_machining_allowance_along_normal

A Casting_machining_allowance_along_normal is a type of Casting_machining_allowance (see 4.2.10) whose surface is obtained by displacing each point of a pre-existing surface by a fixed distance along the normal of the pre-existing surface at the given point. The data associated with a Casting_machining_allowance_along_normal are the following:

- thickness.

The thickness specifies the distance by which each point of the pre-existing surface is to be offset to create the surface of the Casting_machining_allowance_along_normal.

4.2.12 Casting_machining_allowance_along_vector

A Casting_machining_allowance_along_vector is a type of Casting_machining_allowance (see 4.2.10) whose surface is obtained by displacing each point of a pre-existing surface by a fixed distance along a given vector. The data associated with a Casting_machining_allowance_along_vector are the following:

- direction;

- offset_amount.

4.2.12.1 direction

The direction specifies the vector along which the pre-existing surface is to be displaced to create resultant surface.

4.2.12.2 offset_amount

The offset_amount specifies the distance by which to displace each point of the pre-existing surface to create the resultant surface.

4.2.13 Casting_machining_allowance_with_explicit_surface

A Casting_machining_allowance_with_explicit_surface is a type of Casting_machining_allowance (see 4.2.10) that is characterized by an explicitly defined surface. The data associated with a Casting_machining_allowance_with_explicit_surface are the following:

- enveloping_geometry.

The enveloping_geometry specifies the shape of the outer boundary of the machining allowance.

4.2.14 Casting_part

A Casting_part is a specification for a part as produced by the casting process. A casting part contains machining allowances, which will be removed in finishing to produce the customer part.

4.2.15 Casting_round_corner_transition

A Casting_round_corner_transition is a type of Casting_design_feature (see 4.2.9) that is a concave or convex circular arc transition between three or more surfaces that intersect at a vertex. The data associated with a Casting_round_corner_transition are the following:

- radius.

The radius specifies the amount of curvature for the transition between the intersecting surfaces.

4.2.16 Casting_round_edge_transition

A Casting_round_edge_transition is a type of Casting_design_feature (see 4.2.9) that is a concave or convex circular arc transition between two intersecting surfaces. The data associated with a Casting_round_edge_transition are the following:

- radius.

The radius specifies the amount of curvature for the transition between the two surfaces.

4.2.17 Casting_taper

A Casting_taper is a type of Casting_design_feature (see 4.2.9) that is the gradual decrease or increase in width or thickness, applied to an enlogated portion of a casting, that results from adding taper to the pattern or mold used to produce the casting.

NOTES

1 – ISO 8062-1 defines taper as "angle added to a geometrical feature of a pattern or mould to improve product quality or enhance production rates."

2 – In die casting a taper is used to facilitate removal of the casting from the mold. In sand casting a taper is used to facilitate removal of the pattern and rigging assembly from the sand mold. In investment casting a taper is used to help remove the wax or foam pattern from the pattern die.

The data associated with a Casting_taper are the following:

- angle;
- method;
- ratio;
- reference_edge;

– with_respect_to.

4.2.17.1 angle

The angle specifies the draft angle. The angle need not be specified for a particular Casting_taper.

4.2.17.2 method

The method specifies the manner in which the Casting_taper is to be formed. The value shall be "taper+," "taper-," or "taper+-." These terms are defined in ISO 8062-2.

4.2.17.3 ratio

The ratio specifies the declination of the tapered surface with respect to its reference edge. The ratio need not be specified for a particular Casting_taper.

4.2.17.4 reference_edge

The reference_edge specifies the direction which is used to specify the inclination of the taper on a casting surface. If a reference edge is not specified, it is assumed to be perpendicular to the parting_surface.

NOTE – The reference edge is usually a normal to the parting surface, but is some cases this is not possible. One example occurs in die casting, when a slide must be removed in a direction that is not perpendicular to its parting surface.

4.2.17.5 with_respect_to

The with_respect_to specifies the parting surface.

NOTE – In most castings there is only one parting plane or parting surface; however, complex castings may have four or more parting surfaces.

4.2.18 Composition_requirement

A Composition_requirement is a type of Item_requirement (see 4.2.67) that is a specification of a chemical or physical composition which must be satisfied by an item. The data associated with a Composition_requirement are the following:

– alloy_grade;

– made_of.

4.2.18.1 alloy_grade

The alloy_grade specifies the label used to identify the composition of an alloy.

NOTE – Alloy grades may be defined in International or National Standards, and industry or company specifications.

EXAMPLE 3 – AA 356.2, LM25, or Al7Si0.3Mg are labels applied to an aluminum alloy grade which contains nominally 7% Si and 0.3% Mg.

The alloy_grade need not be specified for a particular Composition_requirement.

4.2.18.2 made_of

The made_of specifies the Substance (see 4.2.124) of which an item must be made. If both made_of and alloy_grade are specified, then the item shall meet the combination of requirements of both. The made_of need not be specified for a particular Composition_requirement.

4.2.19 Compound_process

A Compound_process is a type of Process (see 4.2.86) that is decomposed into simpler processes. The data associated with a Compound_process are the following:

– element.

The element specifies the a processing step for a Compound_process.

4.2.20 Computer_file

A Computer_file is a collection of data stored electronically. The data associated with a Computer_file are the following:

– file_name.

The file_name specifies the label used to identify a Computer_file.

4.2.21 Condition_or_assumption

A Condition_or_assumption is a state of physical phenomena or presumption about the state of physical phenomena. The data associated with a Condition_or_assumption are the following:

– associated_properties;

– description;

– further_information.

4.2.21.1 associated_properties

The associated_properties specifies the value of a physical property, or the functional dependence of a physical property, that represents the Condition_or_assumption. The associated_properties need not be specified for a particular Condition_or_assumption.

4.2.21.2 description

The description specifies the textual account of the Condition_or_assumption.

4.2.21.3 further_information

The further_information specifies the additional document that describes or delineates the Condition_or_assumption. The further_information need not be specified for a particular Condition_or_assumption.

4.2.22 Consumable_item

A Consumable_item is a specification for an item which is used up during a Process (see 4.2.86).

EXAMPLE 4 – An ingot that is charged to the furnace and melted for casting is a Consumable_item.

4.2.23 Core

A Core is a design of specially formed material inserted into a mold to shape the interior or other part of a casting that cannot be easily shaped by the pattern. The data associated with a Core are the following:

– construction;

– type_of_core.

4.2.23.1 construction

The construction specifies the design of the core master which is used to construct the core box.

4.2.23.2 type_of_core

The type_of_core specifies the label identifying the kind of core.

EXAMPLE 5 – “Shell cores” and “cold box cores” are kinds of cores.

4.2.24 Core_master

A Core_master is a type of Master (see 4.2.72) that is a reference object that is used to manufacture a pattern or die.

4.2.25 Curve_on_surface

A Curve_on_surface is a curve that lies on a surface.

4.2.26 Curve_point

A Curve_point is location of a data point on a Data_curve (see 4.2.28). The data associated with a Curve_point are the following:

– x_value;

– y_value.

4.2.26.1 x_value

The x_value specifies the numerical measure of the x-coordinate for the Curve_point.

4.2.26.2 y_value

The y_value specifies the numerical measure of the y-coordinate for the Curve_point.

4.2.27 Customer_part

A Customer_part is specification for a part as originally received from the client.

NOTE – The customer may not be aware of, or be concerned with, the limitations of the casting process. Therefore, a Customer_part may need to be modified to enable it to be economically produced by casting.

4.2.28 Data_curve

A Data_curve is a plot that exhibits a relationship between two variables. The data associated with a Data_curve are the following:

– data_point;

– interpolation_method;

– x_unit;

– y_unit.

4.2.28.1 data_point

The data_point specifies the pair of values for two variables through which the Data_curve must pass. There may be more than one data_point for a Data_curve.

4.2.28.2 interpolation_method

The interpolation_method specifies the procedure for estimation of property values in between adjoining sets of data points.

NOTE – Interpolation methods include linear and polynomial interpolation.

4.2.28.3 x_unit

The x_unit specifies the unit of measure for the ordinate of the Data_curve.

4.2.28.4 y_unit

The y_unit specifies the unit of measure for the abscissa of the Data_curve.

4.2.29 Date

A Date is a particular day.

4.2.30 Date_and_time

A Date_and_time is a moment of time on a particular day.

4.2.31 Datum_reference_frame

A Datum_reference_frame is a sequence of datums that constrains the location of a feature on a physical part.

4.2.32 Defect_prediction

A Defect_prediction is an undesirable structural feature that is forecast by a simulation.

EXAMPLE 6 – Possible defects include shrinkage porosity, oxide inclusions, and cracks.

The data associated with a Defect_prediction are the following:

- defect_shape;
- description;
- location;
- type_of_defect.

4.2.32.1 defect_shape

The defect_shape specifies the characteristic spatial form of a defect. The defect_shape need not be specified for a particular Defect_prediction.

4.2.32.2 description

The description specifies the textual account of a Defect_prediction.

4.2.32.3 location

The location specifies the position of a predicted defect within a part. The location need not be specified for a particular Defect_prediction.

4.2.32.4 type_of_defect

The type_of_defect specifies the label that identifies the kind of defect.

NOTE – Shrinkage, porosity, cracks and oxide inclusions are kinds of defects.

4.2.33 Design_request

A Design_request is an appeal for more information about, or a modification to, a design. Each Design_request is either a Request_for_change (see 4.2.98), or a Request_for_clarification (see 4.2.99). The data associated with a Design_request are the following:

- description;
- referenced_item;
- request_date;
- requested_by.

4.2.33.1 description

The description specifies the textual account of a Design_request.

4.2.33.2 referenced_item

The referenced_item specifies the Item (see 4.2.66) that is affected by the Design_request.

4.2.33.3 request_date

The request_date specifies the Date_and_time (see 4.2.30) at which the Design_request is made.

4.2.33.4 requested_by

The requested_by specifies the person making the Design_request.

4.2.34 Design_update

A Design_update is a change or modification made to a design. The data associated with a Design_update are the following:

- action_date;
- action_taker;
- description;
- referenced;
- updated_item.

4.2.34.1 action_date

The action_date specifies the Date_and_time (see 4.2.30) at which a Design_update is made.

4.2.34.2 action_taker

The action_taker specifies the person making the update.

4.2.34.3 description

The description specifies the textual account of the Design_update.

4.2.34.4 referenced

The referenced specifies the Design_request (see 4.2.33) that is addressed by the Design_update.

4.2.34.5 updated_item

The updated_item specifies the Item (see 4.2.66) that is changed by the Design_update.

4.2.35 Die_component

A Die_component is design of a part which forms an element of a die sub-assembly. The data associated with a Die_component are the following:

- type_of_component.

The type_of_component specifies the name of the part which is being described.

EXAMPLE 7 – "Fixed core," "consumable core," "insert," "chill," "insulating sleeve," "in-mold filter" and "ejector pin" are names of die components.

The type_of_component need not be specified for a particular Die_component.

4.2.36 Die_mold_assembly

A Die_mold_assembly is a complete die mechanism used to produce a die casting. The data associated with a Die_mold_assembly are the following:

- cavity_shape.

The cavity_shape specifies the geometry of the empty region contained inside a die mold assembly that will be filled with liquid metal during casting.

NOTE – The cavity shape is an input data for the simulation of the die casting process.

The cavity_shape need not be specified for a particular Die_mold_assembly.

4.2.37 Die_sub_assembly

A Die_sub_assembly is a type of Assembly_item (see 4.2.4) that is the design of an assembly which forms part of a die mold assembly.

NOTE – A Die_sub_assembly may contain other Die_sub_assembly objects. For example, a sub-assembly of slide may contain a sub-assembly of ejector pin and ejector plate, and the die components of the fixed cores.

The data associated with a Die_sub_assembly are the following:

- type_of_sub_assembly.

The type_of_sub_assembly specifies the kind of die sub-assembly.

4.2.38 Dimensional_tolerance

A Dimensional_tolerance is a type of Tolerance (see 4.2.133) that is an amount of variation from a standard or specified value for a structural dimension of an item.

4.2.39 Document

A Document is a written or printed paper bearing information needed at some point of time during casting design or casting manufacture. The data associated with a Document are the following:

- name.

The name specifies the human-understandable term used to identify a Document.

4.2.40 Equipment_usage

An Equipment_usage is a machinery specification for a Process (see 4.2.86). The data associated with an Equipment_usage are the following:

- equipment_name;
- equipment_setting;
- identification;
- machine_program;
- operation_instruction.

4.2.40.1 equipment_name

The equipment_name specifies the label used to identify the equipment.

NOTE – The equipment_name may be the make and model of equipment.

4.2.40.2 equipment_setting

The equipment_setting specifies the machine operating variable for an Equipment_usage.

NOTE – A set point is a kind of machine operating variable.

The equipment_setting need not be specified for a particular Equipment_usage. There may be more than one equipment_setting for a Equipment_usage.

4.2.40.3 identification

The identification specifies the label used to identify a specific piece of machinery. The identification need not be specified for a particular Equipment_usage.

4.2.40.4 machine_program

The machine_program specifies the Computer_file (see 4.2.20) which contains the instructions to control the machine. The machine_program need not be specified for a particular Equipment_usage.

4.2.40.5 operation_instruction

The operation_instruction specifies the text that describes the proper operation of the machine within a Process (see 4.2.86).

4.2.41 Failure_report

A Failure_report is a description of a situation in which a requirement is not met. The data associated with a Failure_report are the following:

- description;
- failed_artifact;
- number_failed;
- what_failed.

4.2.41.1 description

The description specifies the textual account of the Failure_report.

NOTE – The description may include a list of the defects observed and reasons for production of scrap.

4.2.41.2 failed_artifact

The failed_artifact specifies the Artifact (see 4.2.3) which did not meet the requirement. The failed_artifact need not be specified for a particular Failure_report. There may be more than one failed_artifact for a Failure_report.

4.2.41.3 number_failed

The number_failed specifies the quantity of Artifact (see 4.2.3) objects that did not meet the requirements. The number_failed need not be specified for a particular Failure_report.

4.2.41.4 what_failed

The what_failed specifies the Inspection_or_test_result (see 4.2.56) which gives details of what requirements the Artifact (see 4.2.3) did not meet and how.

4.2.42 Feature_parameter

A Feature_parameter is a variable used to characterize a Casting_design_feature (see 4.2.9). The data associated with a Feature_parameter are the following:

- value.

The value specifies the numerical quantity for the variable that characterizes a Casting_design_feature (see 4.2.9).

4.2.43 Finished_casting

A Finished_casting is a type of Simple_item (see 4.2.109) that is a cast part to which the specified finishing operations have been applied.

EXAMPLE 8 – Trimming, grinding, machining and heat treatment are types of finishing operations.

4.2.44 Flask

A Flask is horizontal extent of the flask. The data associated with a Flask are the following:

- cope_height;
- drag_height;
- flask_length;
- flask_width;
- type_of_flask.

4.2.44.1 cope_height

The cope_height specifies the height of the upper portion of the flask.

4.2.44.2 drag_height

The drag_height specifies the height of the lower portion of the flask.

4.2.44.3 flask_length

The flask_length specifies the longitudinal extent of the flask.

4.2.44.4 flask_width

The flask_width specifies the horizontal extent of the flask.

4.2.44.5 type_of_flask

The type_of_flask specifies the kind of flask.

NOTE – The type_of_flask may be a trade name.

4.2.45 Geometric_2d_shape_representation

A Geometric_2d_shape_representation is a type of Shape_representation (see 4.2.108) that is the representation of a shape in which all points fall on a plane.

4.2.46 Geometric_tolerance

A Geometric_tolerance is a type of Tolerance (see 4.2.133) that is an amount of variation allowed for a specific structural dimension, as measured with respect to a datum point or a datum plane. The data associated with a Geometric_tolerance are the following:

- with_respect_to.

The with_respect_to specifies the datum point, or a datum plane, used for the measurement of a specific structural dimension. The with_respect_to need not be specified for a particular Geometric_tolerance.

4.2.47 Geometry_characterization

A Geometry_characterization is the size, shape and geometric orientation of a Substance_structure_element (see 4.2.128). The data associated with a Geometry_characterization are the following:

- characteristic_size;
- description;
- orientation;
- shape.

4.2.47.1 characteristic_size

The characteristic_size specifies the dimensional measure used to describe the physical magnitude of a Substance_structure_element (see 4.2.128).

EXAMPLE 9 – Particle diameter and plate length are dimensional measures.

The characteristic_size need not be specified for a particular Geometry_characterization.

4.2.47.2 description

The description specifies the textual account of the Geometry_characterization.

4.2.47.3 orientation

The orientation specifies the spatial placement of a Substance_structure_element (see 4.2.128).

EXAMPLE 10 – The grain orientation in single crystal jet engine fan blades.

The orientation need not be specified for a particular Geometry_characterization.

4.2.47.4 shape

The shape specifies the typical spatial form of a Substance_structure_element (see 4.2.128). The shape need not be specified for a particular Geometry_characterization.

4.2.48 Geometry_requirement

A Geometry_requirement is a type of Item_requirement (see 4.2.67) that is the specification that an Item (see 4.2.66) have a specific spatial form. The data associated with a Geometry_requirement are the following:

- shape.

The shape specifies the representation of the specific geometry, which has been determined to be necessary for an Item.

4.2.49 Graphics

A Graphics is a marking on a cast part used for identification.

NOTE – A Graphics may be a name, symbol, logo or trademark.

The data associated with a Graphics are the following:

- name.

The name specifies the label used to identify a Graphics.

4.2.50 Heat

A Heat is a batch of metal, obtained from an uninterrupted melting operation in an individual cupola or furnace. The data associated with a Heat are the following:

- heat_number;
- production_date.

4.2.50.1 heat_number

The heat_number specifies the label used to identify a Heat.

4.2.50.2 production_date

The production_date specifies the Date_and_time (see 4.2.30) when the melting of a Heat was completed.

4.2.51 Heat_treat_requirement

A Heat_treat_requirement is a type of Item_requirement (see 4.2.67) that is a sequence of heating and cooling cycles used to obtain desired properties or conditions of an Item (see 4.2.66). The data associated with a Heat_treat_requirement are the following:

- temperature_versus_time.

The temperature_versus_time specifies the Data_curve (see 4.2.28) that specifies the regime of heating and cooling needed to achieve the desired properties or conditions.

4.2.52 Identification_marking

An Identification_marking is a type of Casting_design_feature (see 4.2.9) that is a visible impression on the surface of a cast part by which the part is identified. The data associated with an Identification_marking are the following:

- height;
- marking;
- raised.

4.2.52.1 height

The height specifies the how far the Identification_marking extends above or below the surrounding surface.

4.2.52.2 marking

The marking specifies the Graphics (see 4.2.49) defining the marking text or shape.

4.2.52.3 raised

The raised specifies whether or not Identification_marking extends outward from the surrounding surface.

4.2.53 In_mold_rigging_component

An In_mold_rigging_component is a type of Simple_item (see 4.2.109) that is placed in the mold to modify the flow of metal. The data associated with an In_mold_rigging_component are the following:

- type_of_rigging.

The type_of_rigging specifies the kind of rigging component.

EXAMPLE 11 – Chills, cores, chaplets, insulating sleeves and filters are kinds of rigging components.

4.2.54 Insert

An Insert is a type of Simple_item (see 4.2.109) that is a solid placed into a mold or die prior to pouring the molten metal, so that it will be incorporated into the cast part.

4.2.55 Inspection_or_test_requirement

An Inspection_or_test_requirement is a type of Item_requirement (see 4.2.67) that specifies the examination procedure to check an Item (see 4.2.66) for compliance with a desired condition. The data associated with an Inspection_or_test_requirement are the following:

- frequency;
- inspection_against;
- sampling_run_size;
- test_name.

4.2.55.1 frequency

The frequency specifies the how often the procedure must be performed.

4.2.55.2 inspection_against

The inspection_against specifies the Item_requirement (see 4.2.67) for which the procedure verifies compliance.

4.2.55.3 sampling_run_size

The sampling_run_size specifies the number of castings that must be produced and tested in the sample run prior to production. The sampling_run_size need not be specified for a particular Inspection_or_test_requirement.

4.2.55.4 test_name

The test_name specifies the label used to identify the Inspection_or_test_requirement. The test_name need not be specified for a particular Inspection_or_test_requirement.

4.2.56 Inspection_or_test_result

An Inspection_or_test_result is the outcome of an examination or measurement. The data associated with an Inspection_or_test_result are the following:

- applies_to;

- belongs_to;
- description;
- requirement;
- result;
- result_label.

4.2.56.1 applies_to

The applies_to specifies the location on an item where the inspection was made.

NOTE – Cast parts may be examined for porosity or cracks along a specified surface, where these defects would be problematical.

The applies_to need not be specified for a particular Inspection_or_test_result.

4.2.56.2 belongs_to

The belongs_to specifies the Sampled_set (see 4.2.101) that was inspected.

4.2.56.3 description

The description specifies the textual account of the Inspection_or_test_result.

NOTE – The description may contain an account of porosity or other defects found in a casting, and the results of any mechanical tests made.

4.2.56.4 requirement

The requirement specifies the Item_requirement (see 4.2.67) that was evaluated to produce the Inspection_or_test_result. The requirement need not be specified for a particular Inspection_or_test_result.

4.2.56.5 result

The result specifies the outcome of the Inspection_or_test_result.

4.2.56.6 result_label

The result_label specifies the a short characterization of the Inspection_or_test_result. The result_label need not be specified for a particular Inspection_or_test_result.

4.2.57 Integration_interval

An Integration_interval is a time step employed for numerical integration, and the total time period covered by a portion of the overall simulation run. The data associated with an Integration_interval are the following:

- delta_time;
- length_of_elapsed_time_interval.

4.2.57.1 delta_time

The delta_time specifies the time step employed for a numerical integration step.

4.2.57.2 length_of_elapsed_time_interval

The length_of_elapsed_time_interval specifies the total time period covered by the Integration_interval.

4.2.58 Investment_mold

An Investment_mold is the design of a mold used for investment casting.

NOTE – The Investment_mold includes a description of the mold shape or thickness, and a description of each layer in the case of shell molds.

The data associated with an Investment_mold are the following:

- cavity_shape;
- investment_mold_type.

4.2.58.1 cavity_shape

The cavity_shape specifies the spatial form occupied by the empty cavity contained inside the investment mold.

NOTES

1 – This cavity will be filled with liquid metal during casting.

2 – The cavity shape is needed for the computer simulation of the investment casting process.

The cavity_shape need not be specified for a particular Investment_mold.

4.2.58.2 investment_mold_type

The investment_mold_type specifies the kind of Investment_mold.

NOTE – "Shell mold" and "plaster mold" are kinds of Investment_mold.

4.2.59 Investment_pattern_assembly

A Investment_pattern_assembly is a type of Assembly_item (see 4.2.4) that is the design of an investment pattern tree, which consists of one or more individual patterns mounted onto an investment sprue assembly.

4.2.60 Investment_pattern

A Investment_pattern is a type of Assembly_item (see 4.2.4) that is the design of the consumable pattern that is used to construct the mold in investment casting.

NOTE – Since the investment pattern may contains cores or inserts, it is represented as an assembly item.

The data associated with an Investment_pattern are the following:

– construction.

The construction specifies the pattern die that is used to manufacture the Investment_pattern.

4.2.61 Investment_pattern_die_component

An Investment_pattern_die_component is a type of Simple_item (see 4.2.109) that is used to make an Investment_pattern_die (see 4.2.62).

4.2.62 Investment_pattern_die

An Investment_pattern_die is a type of Assembly_item (see 4.2.4) that is the design of the assembly used by an Investment_pattern (see 4.2.60).

4.2.63 Investment_sprue_assembly

An Investment_sprue_assembly is a type of Assembly_item (see 4.2.4) that forms the piping to facilitate flow of molten metal into an Investment_mold (see 4.2.58).

NOTES

1 – Included are the pour cup, down sprue or sprues, gates, runners and risers and in-mold rigging components, such as filters.

2 – The pour cup, down sprue or sprues, gates, runners and risers are usually mounted together into a single assembly prior to attachment of the investment patterns.

4.2.64 Investment_sprue_component

An Investment_sprue_component is a type of Simple_item (see 4.2.109) that is used to make the sprue assembly in investment casting.

4.2.65 Product

An Product is an object that is produced by, input to, or employed as part of, a Process (see 4.2.86). The data associated with an Product are the following:

- description;
- identification;
- name.

4.2.65.1 description

The description specifies the textual account of an Product.

4.2.65.2 identification

The identification specifies the label used to identify an Product.

4.2.65.3 name

The name specifies the human-understandable term used to identify an Product.

4.2.66 Item

An Item is a physical object which is produced during, or used by, a manufacturing process. Each Item is either an Assembly_item (see 4.2.4), a Consumable_item (see 4.2.22), or a Simple_item (see 4.2.109). The data associated with an Item are the following:

- associated_item_version;
- description;
- designer;
- identification;
- name;
- requirement.

4.2.66.1 associated_item_version

The associated_item_version specifies the which variant of a design is represented by the Item. The associated_item_version need not be specified for a particular Item.

4.2.66.2 description

The description specifies the textual account of the Item.

4.2.66.3 designer

The designer specifies the person responsible for the specification of the Item. The designer need not be specified for a particular Item.

4.2.66.4 identification

The identification specifies the label used to identify the Item. The identification need not be specified for a particular Item.

4.2.66.5 name

The name specifies the human-understandable term used to identify the Item.

4.2.66.6 requirement

The requirement specifies the constraints that must be satisfied by the Item. There may be more than one requirement for a Item.

4.2.67 Item_requirement

An Item_requirement is a constraint that an Item (see 4.2.66) must satisfy. Each Item_requirement may be one of the following: a Composition_requirement (see 4.2.18), a Geometry_requirement (see 4.2.48), a Heat_treat_requirement (see 4.2.51), an Inspection_or_test_requirement (see 4.2.55), a Process_requirement (see 4.2.92), a Property_requirement (see 4.2.94), a Reporting_requirement (see 4.2.97), a Required_machining_allowance (see 4.2.100), or a Surface_roughness_requirement (see 4.2.131). The data associated with an Item_requirement are the following:

- applies_to;
- description;
- illustration;
- referenced_from.

4.2.67.1 applies_to

The applies_to specifies the portion or surface of an Item (see 4.2.66) that satisfies the Item_-requirement.

EXAMPLE 12 – A certain hardness may be specified for a piston pin sleeve.

The applies_to need not be specified for a particular Item_requirement.

4.2.67.2 description

The description specifies the textual account of an Item_requirement.

4.2.67.3 illustration

The illustration specifies the an image that illustrates the Item_requirement. The illustration need not be specified for a particular Item_requirement.

4.2.67.4 referenced_from

The referenced_from specifies the specifications upon which the Item_requirement is based. The referenced_from need not be specified for a particular Item_requirement.

4.2.68 Item_version

An Item_version is a type of Version (see 4.2.137) that identifies a variant of an Item (see 4.2.66) that has undergone a formal release or change.

NOTE – Iterations on designs that are not formally tracked by an organization responsible for the definition of a product are tracked using Item (see 4.2.66) .

The data associated with an Item_version are the following:

- associated_item;
- purchase_order_numbers.

4.2.68.1 associated_item

The associated_item specifies the Product (see 4.2.65) for which an Item_version is a variant.

4.2.68.2 purchase_order_number

The purchase_order_number specifies the label used to identify a purchase order for this Item_version.

4.2.69 Lot

A Lot is a group of Artifact (see 4.2.3) objects that are manufactured together or in a sequence.

NOTE – A lot may be established by the foundry; for example, to record the day when a group of castings are made. A lot may also be specified by the customer; for example, when a lot of 500 castings are specified for delivery monthly.

The data associated with a Lot are the following:

- lot_number;
- production_date.

4.2.69.1 lot_number

The lot_number specifies the numerical label used to identify the Lot.

4.2.69.2 production_date

The production_date specifies the Date_and_time (see 4.2.30) when the lot was manufactured, or when production of the lot was completed.

4.2.70 Machine_setting

A Machine_setting is a value for an equipment operating variable to be maintained as part of a Process_plan (see 4.2.90). The data associated with a Machine_setting are the following:

- parameter_name;
- parameter_setting.

4.2.70.1 parameter_name

The parameter_name specifies the label used to identify a Machine_setting variable.

EXAMPLE 13 – In die casting, pressure at end of fill and intensification pressure are possible parameter names.

4.2.70.2 parameter_setting

The parameter_setting specifies the value for a Machine_setting.

4.2.71 Machine_setting_record

A Machine_setting_record is a record of a value for an equipment operating variable during a particular execution of a Process (see 4.2.86).

NOTE – A Machine_setting_record may be used for later analysis to determine the effect of an equipment operating variable on the quality of the part.

The data associated with a Machine_setting_record are the following:

- actual_setting;
- equipment_usage_context.

4.2.71.1 actual_setting

The actual_setting specifies the measured value for a Machine_setting (see 4.2.70).

4.2.71.2 equipment_usage_context

The equipment_usage_context specifies the Equipment_usage (see 4.2.40) for which the value of an equipment operating variable was measured.

4.2.72 Master

A Master is a type of Simple_item (see 4.2.109) that is used to manufacture cores or patterns. Each Master is either a Pattern_master (see 4.2.81), or a Core_master (see 4.2.24). The data associated with a Master are the following:

- storage_location.

The storage_location specifies the a label or human-readable text indicating where the master is placed when it is not in use. The equipment_usage_context need not be specified for a particular Master.

4.2.73 Measure_range_with_unit

A Measure_range_with_unit is a range of values and a desired nominal value specified for a physical characteristic or property. The data associated with a Measure_range_with_unit are the following:

- lower_limit;
- nominal_amount;
- upper_limit.

4.2.73.1 lower_limit

The lower_limit specifies the lowest permissible value specified for the physical property. The lower_limit need not be specified for a particular Measure_range_with_unit.

4.2.73.2 nominal_amount

The nominal_amount specifies the planned or designed value for the physical property. The nominal_amount need not be specified for a particular Measure_range_with_unit.

4.2.73.3 upper_limit

The upper_limit specifies the largest permissible value for the physical property. The upper_limit need not be specified for a particular Measure_range_with_unit.

4.2.74 Measure_with_unit

A Measure_with_unit is a combination of a numerical quantity with dimensions or units, used to represent physical characteristics or properties.

NOTE – Physical characteristcs or properties can be measured experimentally or empirically.

EXAMPLE 14 – For temperature, "1400 degrees Celsius"; for pressure, "1500 millibars"; for time, "20.5 seconds."

4.2.75 Meshing_condition

A Meshing_condition is a description of the fineness of mesh needed in a region. The data associated with a Meshing_condition are the following:

– areas_to_be_meshed;

– density_of_elements;

– density_of_nodes;

– number_of_elements;

– number_of_nodes.

4.2.75.1 areas_to_be_meshed

The areas_to_be_meshed specifies the region to which the Meshing_condition applies.

NOTE – A Meshing_condition may apply to an entire Item (see 4.2.66) , or a portion of an Item.

The areas_to_be_meshed need not be specified for a particular Meshing_condition.

4.2.75.2 density_of_elements

The density_of_elements specifies the number of elements per unit volume for the region. The density_of_elements need not be specified for a particular Meshing_condition.

4.2.75.3 density_of_nodes

The density_of_nodes specifies the number of nodes per unit volume for the region. The density_of_nodes need not be specified for a particular Meshing_condition.

4.2.75.4 number_of_elements

The number_of_elements specifies the quantity of elements for region.

4.2.75.5 number_of_nodes

The number_of_nodes specifies the quantity of nodes for region.

4.2.76 Mismatch_tolerance

A Mismatch_tolerance is a type of Tolerance (see 4.2.133) that is an imperfection on a cast part caused by inaccurate alignment or displacement or dimensional differences between the mold or die components, producing a step on the surface along the parting line. The data associated with a Mismatch_tolerance are the following:

- maximum_amount;
- with_respect_to.

4.2.76.1 maximum_amount

The maximum_amount specifies the largest allowable amount of mismatch.

4.2.76.2 with_respect_to

The with_respect_to specifies the Parting_surface (see 4.2.78) with which the Mismatch_tolerance is associated.

NOTE – In castings which have more than one parting surface, it may be desirable to specify a different amount of mismatch along each surface.

The with_respect_to need not be specified for a particular Mismatch_tolerance.

4.2.77 Organization

An Organization is an administrative structure.

4.2.78 Parting_surface

A Parting_surface is a layer of separation between adjoining sections of a mold. The data associated with a Parting_surface are the following:

- surface_geometry.

The surface_geometry specifies the shape of the Parting_surface.

NOTE – Normally a Parting_surface is a planar surface, but it may be curved, or consist of several planar sections joined at angles.

4.2.79 Pattern_and_rigging

A Pattern_and_rigging is a type of Assembly_item (see 4.2.4) that is the complete assembly which, when pressed into the sand, forms the void into which molten metal is introduced as part of the sand casting process.

NOTE – The pattern and rigging assembly consists of the pattern plate, one or more patterns, and a number of rigging objects. Except in the case of a loose pattern, the patterns and rigging objects are attached in a fixed position onto both sides of the pattern plate.

4.2.80 Pattern

A Pattern is a type of Simple_item (see 4.2.109) that is the same shape as the raw casting.

NOTE – A Pattern may be permanent or expendable. In sand casting, the pattern is used to form the cavity for the part shape in the sand mold. In investment casting, the investment is formed around the pattern, then the pattern is melted away to leave the desired cavity shape. In die casting, the pattern serves as a guide for cutting the proper cavity shape in the die.

The data associated with a Pattern are the following:

– construction.

The construction specifies the Pattern_master (see 4.2.81) that is used to manufacture the Pattern .

4.2.81 Pattern_master

A Pattern_master is a type of Master (see 4.2.72) that is the same shape as a Pattern (see 4.2.80), and is replicated to make one or more Pattern (see 4.2.80) objects.

4.2.82 Pattern_plate

A Pattern_plate is a type of Simple_item (see 4.2.109) to which the pattern and rigging are mounted for use in the sand casting process.

NOTE – A Pattern_plate is usually a flat, rectangular plate made of wood or metal.

The data associated with a Pattern_plate are the following:

– type_of_plate.

The type_of_plate specifies the kind of Pattern_plate.

EXAMPLE 15 – “Pressure plate” and “match plate” are two kinds of pattern plates.

4.2.83 Pattern_rigging_component

A Pattern_rigging_component is a type of Simple_item (see 4.2.109) that is used to form a part of the plumbing system in a mold, through which molten metal flows during casting, and which is not within the part cavity. The data associated with a Pattern_rigging_component are the following:

– type_of_rigging.

The type_of_rigging specifies the kind of Pattern_rigging_component.

EXAMPLE 16 – “Down sprue,” “gate” and “runner” are kinds of rigging components.

4.2.84 Person_and_organization

A Person_and_organization is an individual, along with the group that he or she represents.

4.2.85 Picture

A Picture is a visual representation or image, drawn, photographed, or otherwise rendered onto a flat surface. The data associated with a Picture are the following:

- title.

The title specifies the label used to identify the Picture.

4.2.86 Process

A Process is a series of actions directed toward changing a part from one state to another. Each Process is either a Compound_process (see 4.2.19), or a Unit_process (see 4.2.134). The data associated with a Process are the following:

- description;
- equipment_used;
- identification;
- illustration;
- input_item;
- name;
- output_item;
- process_parameter;
- reference_to.

4.2.86.1 description

The description specifies the textual account of the Process.

4.2.86.2 equipment_used

The equipment_used specifies the machine used by the Process. The equipment_used need not be specified for a particular Process.

4.2.86.3 identification

The identification specifies the label used to identify the Process. The identification need not be specified for a particular Process.

4.2.86.4 illustration

The illustration specifies the image that shows detail about the Process.

EXAMPLES

17 – A photograph used to show the location of patterns on a pattern tree in investment casting.

18 – A sketch used to show where inspections for cracks or porosity are to be carried out in a large casting.

The illustration need not be specified for a particular Process.

4.2.86.5 input_item

The input_item specifies the object that the Process transforms, or consumes. The input_item need not be specified for a particular Process.

4.2.86.6 name

The name specifies the designation of the Process.

4.2.86.7 output_item

The output_item specifies the object produced by the Process. The output_item need not be specified for a particular Process.

4.2.86.8 process_parameter

The process_parameter specifies the operating variables for the Process.

EXAMPLE 19 – "Mold temperature" and "tilt rate" are operating variables in low pressure die casting.

The process_parameter need not be specified for a particular Process.

4.2.86.9 reference_to

The reference_to specifies the specification or standard specifying the Process.

EXAMPLE 20 – The schedule of heat treatments established for aluminum-based alloys by the Aluminum Association (United States).

The reference_to need not be specified for a particular Process.

4.2.87 Process_execution

A Process_execution is the performance of a Process_plan (see 4.2.90) to manufacture a cast part.

NOTE – The Process_execution object may be used to record differences between planned and actual conditions, as well as specific conditions existed, when a range of conditions is allowed by the Process_plan (see 4.2.90).

The data associated with a Process_execution are the following:

– description;

– deviated_process;

– parts_measured;

– plan_used.

4.2.87.1 description

The description specifies the written account of the Process_execution.

4.2.87.2 deviated_process

The deviated_process specifies the actual series of actions by which a cast part was made.

4.2.87.3 parts_measured

The parts_measured specifies the Sampled_set (see 4.2.101) that measures the properties of the parts produced by the Process_execution. The parts_measured need not be specified for a particular Process_execution.

4.2.87.4 plan_used

The plan_used specifies the Process_plan (see 4.2.90) that was followed during the Process_execution , possibly with some deviations.

4.2.88 Process_execution_record

A Process_execution_record is a log of how a manufacturing operation was carried out.

NOTE – Process_execution_record objects, in combination with inspection results, can be used by techniques such as statistical process control, to determine the relationship between quality factors and process variables, thus enabling continuous improvement of product quality.

Each Process_execution_record is either a Machine_setting_record (see 4.2.71), a Process_parameter_record (see 4.2.89), a Substance_composition_element_record (see 4.2.126), or a Substance_usage_record (see 4.2.130). The data associated with a Process_execution_record are the following:

- process_context;
- recording_date.

4.2.88.1 process_context

The process_context specifies the Process (see 4.2.86) that was being executed when the Process_execution_record was made.

4.2.88.2 recording_date

The recording_date specifies the day and time at which the Process_execution_record was made.

NOTE – This would normally be when the actual process parameters are measured and recorded during production.

The recording_date need not be specified for a particular Process_execution_record.

4.2.89 Process_parameter_record

A Process_parameter_record is a type of Process_execution_record (see 4.2.88) a log of the value of a manufacturing variable during manufacture of a part. The data associated with a Process_parameter_record are the following:

- actual_parameter;
- planned_parameter.

4.2.89.1 actual_parameter

The actual_parameter specifies the measured value of a manufacturing variable.

4.2.89.2 planned_parameter

The planned_parameter specifies the expected value of a manufacturing variable.

4.2.90 Process_plan

A Process_plan is a set of instructions for making a part. The data associated with a Process_plan are the following:

- description;
- designed_date;
- designer;

- identification;
- name;
- operation.

4.2.90.1 description

The description specifies the textual account of the Process_plan.

4.2.90.2 designed_date

The designed_date specifies the Date_and_time (see 4.2.30) of creation for the Process_plan.

4.2.90.3 designer

The designer specifies the person responsible for creating the Process_plan. The designer need not be specified for a particular Process_plan.

4.2.90.4 identification

The identification specifies the label used to identify the Process_plan.

4.2.90.5 name

The name specifies the human-understandable text used to identify the Process_plan.

4.2.90.6 operation

The operation specifies the step that constitutes the process plan. There may be more than one operation for a Process_plan.

4.2.91 Process_plan_version

A Process_plan_version is a type of Version (see 4.2.137) a variant of a Process_plan (see 4.2.90) The data associated with a Process_plan_version are the following:

- associated_plan.

The associated_plan specifies the Process_plan (see 4.2.90) one variant of which is given by this Process_plan_version .

4.2.92 Process_requirement

A Process_requirement is a type of Item_requirement (see 4.2.67) that specifies that an action must be performed on a part during manufacture.

EXAMPLE 21 – Heat treatment, various surface finishing operations, and various cleaning operations are examples.

The data associated with a Process_requirement are the following:

- process_name.

The process_name specifies the human-understandable term used to identify the required action.

4.2.93 Property_relationship

A Property_relationship is the allowable pairs of values for two variables.

EXAMPLE 22 – A plot of heat treating oven temperature versus time.

The data associated with a Property_relationship are the following:

- relationship_curve;
- x_property_name;
- y_property_name.

4.2.93.1 relationship_curve

The relationship_curve specifies the x-y plot that is used to describe the correlation between the two variables.

4.2.93.2 x_property_name

The x_property_name specifies the label used to identify the first property.

4.2.93.3 y_property_name

The y_property_name specifies the label used to identify the second property.

4.2.94 Property_requirement

A Property_requirement is a type of Item_requirement (see 4.2.67) that specifies a condition that a variable must meet. The data associated with a Property_requirement are the following:

- definition.

The definition specifies the description of the variable for which the Property_requirement sets a condition.

4.2.95 Property_value

A Property_value is the condition that a variable is associated with a specific numerical quantity. The data associated with a Property_value are the following:

- name;
- value.

4.2.95.1 name

The name specifies the label used to identify the variable.

4.2.95.2 value

The value specifies the numerical quantity characterizes the state of the variable.

4.2.96 Raw_casting

A Raw_casting is a type of Simple_item (see 4.2.109) that is a casting just after it has been removed from the mold.

4.2.97 Reporting_requirement

A Reporting_requirement is a type of Item_requirement (see 4.2.67) that specifies information that must be disclosed. The data associated with a Reporting_requirement are the following:

- based_on;
- data_to_be_reported.

4.2.97.1 based_on

The based_on specifies the Item_requirement (see 4.2.67) that is verified by the Reporting_requirement .

4.2.97.2 data_to_be_reported

The data_to_be_reported specifies the informational item that is to be furnished to satisfy the Reporting_requirement .

4.2.98 Request_for_change

A Request_for_change is a type of Design_request (see 4.2.33) that is an appeal for a modification to be made in a design. The data associated with a Request_for_change are the following:

– suggested_change.

The suggested_change specifies the change that is being proposed by the Request_for_change.

4.2.99 Request_for_clarification

A Request_for_clarification is a type of Design_request (see 4.2.33) an appeal for missing information, or additional information to eliminate possible sources of confusion or error concerning a design.

EXAMPLES

23 – A dimension that is not supplied.

24 – Dimensions that do not add up.

4.2.100 Required_machining_allowance

A Required_machining_allowance is a type of Item_requirement (see 4.2.67) defined in ISO 8062-1. The data associated with a Required_machining_allowance are the following:

– value.

The value specifies the numerical quantity of material thickness that characterizes the Required_machining_allowance .

4.2.101 Sampled_set

A Sampled_set is a group of parts that are inspected or tested together.

EXAMPLE 25 – In a particular process, every tenth casting is tested or examined thoroughly for a hardness, the value of a critical dimension, and other specified properties. The results of the inspection are presented daily for statistical process control. In this case the sample size is 10 per cent of the day's production.

The data associated with a Sampled_set are the following:

– artifacts_inspected;

– sampled_from;

– size.

4.2.101.1 artifact_inspected

The artifact_inspected specifies the item that has been inspected. The artifact_inspected need not be specified for a particular Sampled_set.

4.2.101.2 sampled_from

The sampled_from specifies the Heat (see 4.2.50) or Lot (see 4.2.69) from which the item has been selected.

NOTE – The sample set of artifacts (usually castings) has been produced by removing a certain number of artifacts from a production lot, or from a production run made with a specific heat.

4.2.101.3 size

The size specifies the number of items in the Sampled_set. The size need not be specified for a particular Sampled_set.

4.2.102 Sand_mold_assembly

A Sand_mold_assembly is a type of Assembly_item (see 4.2.4) two or more sand molds that are joined together to produce a larger assembly, into which molten metal is poured.

EXAMPLE 26 – A sand mold assembly is a mold which also has a smaller, secondary mold attached, which is known as a “cheek” mold.

4.2.103 Sand_mold_component

A Sand_mold_component is a type of Simple_item (see 4.2.109) that forms a solid portion of a Sand_mold (see 4.2.104). The data associated with a Sand_mold_component are the following:

– construction_flask;

– construction_pattern_and_rigging.

4.2.103.1 construction_flask

The construction_flask specifies the Flask (see 4.2.44) that is used to contain the sand during manufacture of the Sand_mold_component.

NOTE – The inside dimensions of the flask determine the outer dimensions of the sand mold.

4.2.103.2 construction_pattern_and_rigging

The construction_pattern_and_rigging specifies the Pattern_and_rigging (see 4.2.79) that, during manufacture of the Sand_mold (see 4.2.104), prevents sand from filling the region where molten metal will flow during pouring.

4.2.104 Sand_mold

A Sand_mold is a type of Assembly_item (see 4.2.4) into which liquid metal is poured and allowed to solidify to form a cast part.

NOTES

1 – Normally the sand mold consists of two halves (the cope and drag portions of the mold), but in some cases there is more than one parting line and the sand mold definition is more complex.

2 – Items such as cores, chills, chaplets, insulating sleeves, in-mold filters, or in mold additions of flux, grain refiner, or innoculant, may be placed in a Sand_mold.

The data associated with a Sand_mold are the following:

– cavity_shape.

The cavity_shape specifies the empty region in the Sand_mold into which liquid metal is poured.

NOTE – The cavity shape is needed during simulation.

The cavity_shape need not be specified for a particular Sand_mold.

4.2.105 Scalar_measure

A Scalar_measure is a value that can be characterized by a single numerical quantity. The data associated with a Scalar_measure are the following:

– scalar.

The scalar specifies the a physical quantity that has a magnitude but not an orientation.

4.2.106 Secondary_tooling

A Secondary_tooling is supplementary devices used to facilitate the casting process. The data associated with a Secondary_tooling are the following:

– type_of_tooling.

The type_of_tooling specifies the kind of Secondary_tooling.

EXAMPLE 27 – "Extraction gripper", "trim press", and "grinding fixture" are kinds of Secondary_-tooling .

4.2.107 Shape_aspect

A Shape_aspect is a region of interest with respect to the shape of an object.

4.2.108 Shape_representation

A Shape_representation is the definition of the shape of an object. Each Shape_representation is either a B_rep_shape_representation (see 4.2.7), or a Geometric_2d_shape_representation (see 4.2.45).

4.2.109 Simple_item

A Simple_item is a type of Item (see 4.2.66) that is a discrete part.

4.2.110 Simulated_property

A Simulated_property is a physical variable whose value is predicted by a simulation.

EXAMPLE 28 – Temperature, fraction liquid, velocity, and state of stress.

The data associated with a Simulated_property are the following:

- name;
- value.

4.2.110.1 name

The name specifies the label used to identify the Simulated_property.

4.2.110.2 value

The value specifies the numerical quantity that characterizes the Simulated_property.

4.2.111 Simulation_input

A Simulation_input is the collection of data required by simulation software in order to begin a Simulation_run (see 4.2.116) . The data associated with an assumption are the following:

- condition;
- input_parameter;
- integration_time_specification;
- region;
- setup_file.

4.2.111.1 assumption

The assumption specifies the presumption about the state of physical phenomena for a Simulation_input . The assumption need not be specified for a particular Simulation_input. There may be more than one assumption for a Simulation_input.

4.2.111.2 condition

The condition specifies the constraints on physical phenomena for a Simulation_input . The condition need not be specified for a particular Simulation_input. There may be more than one condition for a Simulation_input.

4.2.111.3 input_parameter

The input_parameter specifies the values for a Simulation_input . There may be more than one input_parameter for a Simulation_input.

4.2.111.4 region

The region specifies the particular section of the Simulation_input . There may be more than one region for a Simulation_input.

4.2.111.5 integration_time_specification

The integration_time_specification specifies the time factor need for a Simulation_input . There may be more than one integration_time_specification for a Simulation_input.

4.2.111.6 setup_file

The setup_file specifies the identifies a document that holds the Simulation_input . The setup_file need not be specified for a particular Simulation_input.

4.2.112 Simulation_input_region

A Simulation_input_region is a portion of space containing similar material properties at the beginning of a Simulation_run (see 4.2.116). The data associated with a simulation_input_region are the following:

- associated_item;
- boundary_condition;
- meshing_condition;
- region_property.

4.2.112.1 associated_item

The associated_item specifies the Item (see 4.2.66) objects which make up the Simulation_input_region . The associated_item need not be specified for a particular Simulation_input_region. There may be more than one associated_item for a Simulation_input_region.

4.2.112.2 boundary_condition

The boundary_condition specifies the constraints on physical phenomena at the border of the Simulation_input_region . The boundary_condition need not be specified for a particular Simulation_input_region. There may be more than one boundary_condition for a Simulation_input_region.

4.2.112.3 meshing_condition

The meshing_condition specifies the rules for dividing the Simulation_input_region into a mesh for simulation. The meshing_condition need not be specified for a particular Simulation_input_region. There may be more than one meshing_condition for a Simulation_input_region.

4.2.112.4 region_property

The region_property specifies the value of a physical variable or a relationship between physical variables for the Simulation_input_region. There may be more than one region_property for a Simulation_input_region.

4.2.112.5 region_type

The region_type specifies the kind of Simulation_input_region.

EXAMPLE 29 – "Empty cavity," "liquid metal," "cores," "mold," "mold coating," and "filter" are region types.

4.2.112.6 region_shape

The region_shape specifies the spatial form of the Simulation_input_region.

4.2.113 Simulation_output

A Simulation_output is the data produced by a simulation software as the result of a Simulation_run (see 4.2.116) . The data associated with a Simulation_output are the following:

- predicted_defect;
- result.

4.2.113.1 predicted_defect

The predicted_defect specifies the an undesirable feature that the Simulation_run (see 4.2.116) indicates will occur. The predicted_defect need not be specified for a particular Simulation_output. There may be more than one predicted_defect for a Simulation_output.

4.2.113.2 result

The result specifies the a specific outcome of a Simulation_run (see 4.2.116). The result need not be specified for a particular Simulation_output. There may be more than one result for a Simulation_output.

4.2.114 Simulation_output_region

A Simulation_output_region is a portion of space containing similar material properties at the end of a Simulation_run (see 4.2.116). The data associated with a Simulation_output_region are the following:

- belongs_to;
- history;
- region_shape.

4.2.114.1 belongs_to

The belongs_to specifies the Simulation_input_region (see 4.2.112) of which the Simulation_output_region is a subregion.

4.2.114.2 region_shape

The region_shape specifies the spatial form of the Simulation_output_region.

4.2.114.3 history

The history specifies the predicted property values at different times during the simulation.

4.2.115 Simulation_result

A Simulation_result is a specific outcome during a specific time period of a Simulation_run (see 4.2.116) . The data associated with a Simulation_result are the following:

- begin_time;
- end_time;
- region;
- illustration.

4.2.115.1 begin_time

The begin_time specifies the first simulation time reported by the Simulation_result.

4.2.115.2 end_time

The end_time specifies the last simulation time reported by the Simulation_result.

4.2.115.3 region

The region specifies the Simulation_output_region (see 4.2.114) in which the Simulation_result occurred. There may be more than one region for a Simulation_result.

4.2.115.4 illustration

The illustration specifies the picture that contains the output of a Simulation_result . The illustration need not be specified for a particular Simulation_result. There may be more than one illustration for a Simulation_result.

4.2.116 Simulation_run

A Simulation_run is an execution of a simulation software program to model a casting process. The data associated with a Simulation_run are the following:

- associated_process;
- description;
- identification;
- input;
- output;
- software_used.

4.2.116.1 associated_process

The associated_process specifies the Process (see 4.2.86) modelled by the Simulation_run . The associated_process need not be specified for a particular Simulation_run.

4.2.116.2 description

The description specifies the textual account of the Simulation_run. There may be more than one description for a Simulation_run.

4.2.116.3 identification

The identification specifies the label used to identify a Simulation_run.

4.2.116.4 input

The input specifies the data used by the simulation software to begin a Simulation_run .

4.2.116.5 output

The output specifies the data produced by the simulation software as the result of a Simulation_-run .

4.2.116.6 software_used

The software_used specifies the computer program used to model the casting process.

4.2.117 Simulation_run_relationship

A Simulation_run_relationship is an association between two different Simulation_run (see 4.2.116) objects.

NOTE – The meaning of the relationship is given by the description attribute.

The data associated with a Simulation_run_relationship are the following:

- description;
- related;
- relating.

4.2.117.1 description

The description specifies the textual account of the Simulation_run_relationship.

4.2.117.2 related

The related specifies the first Simulation_run (see 4.2.116) in the relationship.

4.2.117.3 relating

The relating specifies the second Simulation_run (see 4.2.116) in the relationship.

4.2.118 Simulation_software

A Simulation_software is the computer program used to model the casting process.

NOTE – The Simulation_software mimics the behaviour of the casting process and shows values of various process variables at various times.

The data associated with a Simulation_software are the following:

- release_date;
- software_name;

– software_version.

4.2.118.1 release_date

The release_date specifies the Date (see 4.2.29) on which the Simulation_software was made available for use. The release_date need not be specified for a particular Simulation_software.

4.2.118.2 software_name

The software_name specifies the label used to identify the Simulation_software.

4.2.118.3 software_version

The software_version specifies the variant of the Simulation_software used for the simulation. The software_version need not be specified for a particular Simulation_software.

4.2.119 Simulation_unit

A Simulation_unit is a point or region which is not broken down further for the purpose of simulation. The data associated with a Simulation_unit are the following:

– belongs_to;

– history;

– name;

– unit_location;

– unit_shape.

4.2.119.1 belongs_to

The belongs_to specifies the Simulation_output_region (see 4.2.114) containing the Simulation_unit .

4.2.119.2 history

The history specifies the value of a simulated property as a function of simulation time. There may be more than one history for a Simulation_unit.

4.2.119.3 name

The name specifies the label used to identify the Simulation_unit.

4.2.119.4 unit_location

The unit_location specifies the placement of the Simulation_unit with respect the simulation coordinate system.

4.2.119.5 unit_shape

The unit_shape specifies the spational form occupied by the Simulation_unit. The unit_shape need not be specified for a particular Simulation_unit.

4.2.120 Simulation_unit_state

A Simulation_unit_state is the condition of the material inside a Simulation_unit (see 4.2.119) at a particular value of simulation time. The data associated with a Simulation_unit_state are the following:

- evaluation_time;
- property.

4.2.120.1 evaluation_time

The evaluation_time specifies the simulation time at which property values were sampled.

4.2.120.2 property

The property specifies the physical value that was computed at the evaluation_time for a Simulation_unit (see 4.2.119) . There may be more than one property for a Simulation_unit_state.

4.2.121 Specification

A Specification is a document containing an exact statement of rules, procedures or methods to be followed.

EXAMPLE 30 – An International Standard, a national standard, or a company document.

The data associated with a Specification are the following:

- issuing_organization;
- specification_identification.

4.2.121.1 issuing_organization

The issuing_organization specifies the Organization (see 4.2.77) that provides the Specification . There may be more than one issuing_organization for a Specification.

4.2.121.2 specification_identification

The specification_identification specifies the label used to identify the Specification. There may be more than one specification_identification for a Specification.

4.2.122 Specification_reference

A Specification_reference is a citation of an item within a Specification (see 4.2.121).

EXAMPLE 31 – A part is to be made of aluminum. The material XXXX is chosen, but it is found that the EL content must be less than YY. The designation is "4014," the referenced_specifcation is the Aluminum Handbook, and the exception is "EL less than YY." COMPLETE THIS.

The data associated with a Specification_reference are the following:

– designation;

– exception;

– referenced_specification.

4.2.122.1 designation

The designation specifies the label used to identify an item within the Specification (see 4.2.121). The designation need not be specified for a particular Specification_reference.

4.2.122.2 exception

The exception specifies the label used to identify the Specification (see 4.2.121). The exception need not be specified for a particular Specification_reference. There may be more than one exception for a Specification_reference.

4.2.122.3 referenced_specification

The referenced_specification specifies the Specification (see 4.2.121) being cited. The referenced_specification need not be specified for a particular Specification_reference. There may be more than one referenced_specification for a Specification_reference.

4.2.123 Statistical_measure

A Statistical_measure is an estimate of a parameter obtained from sampling. The data associated with a Statistical_measure are the following:

– mean;

– number_of_measurement;

– variance.

4.2.123.1 mean

The mean specifies the simple algebraic mean of the measured values of the population.

4.2.123.2 number_of_measurement

The number_of_measurement specifies the size of the population.

4.2.123.3 variance

The variance specifies the normal mean square deviation of the measured values of the population.

4.2.124 Substance

A Substance is a tangible material that is produced or consumed by a Process (see 4.2.86), or that is needed to facilitate a Process (see 4.2.86). The data associated with a Substance are the following:

- composition;
- description;
- physical_form;
- structure.

4.2.124.1 composition

The composition specifies the actual material content of a Substance . The composition need not be specified for a particular Substance. There may be more than one composition for a Substance.

4.2.124.2 description

The description specifies the written account of the Substance. The description need not be specified for a particular Substance.

4.2.124.3 physical_form

The physical_form specifies the material state of the Substance.

EXAMPLE 32 – Solid, liquid, and gas are physical forms.

The physical_form need not be specified for a particular Substance. There may be more than one physical_form for a Substance.

4.2.124.4 structure

The structure specifies the composition of the Substance in terms of smaller physical units. The structure need not be specified for a particular Substance.

4.2.125 Substance_composition_element

A Substance_composition_element is the amount of an element in a Substance (see 4.2.124). The data associated with a Substance_composition_element are the following:

- amount;
- element.

4.2.125.1 amount

The amount specifies the quantative measure of a chemical element in a Substance (see 4.2.124).

4.2.125.2 element

The element specifies the chemical element that is contained in a Substance (see 4.2.124) in the amount specified by the amount attribute.

4.2.126 Substance_composition_element_record

A Substance_composition_element_record is a type of Process_execution_record (see 4.2.88) a record of the quantative measure of a chemical element in a Substance (see 4.2.124) during a particular execution of a Process (see 4.2.86).

NOTE – A Substance_composition_element_record may be used for later analysis to determine the effect of chemical composition on the quality of the part.

The data associated with a Substance_composition_element_record are the following:

- actual_substance_composition_amount;
- substance_usage_context.

4.2.126.1 actual_substance_composition_amount

The actual_substance_composition_amount specifies the measured amount of a chemical element in a part during a particular execution of a process.

4.2.126.2 substance_usage_context

The substance_usage_context specifies the Substance_usage (see 4.2.129) within a process whose actual value is recorded by the Substance_composition_element_record.

4.2.127 Substance_structure

A Substance_structure is the representation of the physical characteristics of a Substance (see 4.2.124). The data associated with a Substance_structure are the following:

- element;
- illustration;
- reference_to.

4.2.127.1 element

The element specifies the chemical element that is constained in a Substance_structure_element (see 4.2.128). There may be more than one element for a Substance_structure.

4.2.127.2 illustration

The illustration specifies the an image that illustrates the Substance_structure. The illustration need not be specified for a particular Substance_structure.

4.2.127.3 reference_to

The reference_to specifies the specification or standard specifying the Substance_structure. The reference_to need not be specified for a particular Substance_structure.

4.2.128 Substance_structure_element

A Substance_structure_element is a physical constitutent of a Substance (see 4.2.124). The data associated with a Substance_structure_element are the following:

- amount;
- characteristic_geometry;
- made_of;
- type_of_element.

4.2.128.1 amount

The amount specifies the quantitiatve measure of the physical constitutent.

4.2.128.2 characteristic_geometry

The characteristic_geometry specifies the size, shape and orientation of the physical constitutent. The characteristic_geometry need not be specified for a particular Substance_structure_element.

4.2.128.3 made_of

The made_of specifies the substance comprising the physical constitutent.

4.2.128.4 type_of_element

The type_of_element specifies the kind of Substance_structure_element.

4.2.129 Substance_usage

A Substance_usage is the kind and amount of material used in a Process (see 4.2.86). The data associated with a Substance_usage are the following:

- amount;
- made_of.

4.2.129.1 amount

The amount specifies the quantitative measure of material to be employed.

4.2.129.2 made_of

The made_of specifies the the material to be employed.

4.2.130 Substance_usage_record

A Substance_usage_record is a log of material used during a Process (see 4.2.86).

NOTE – A Substance_usage_record may be used for later inspection and analysis to determine the relationship between manufacturing variables and product quality.

The data associated with a Substance_usage_record are the following:

- actual_substance_usage_amount.

The amount specifies the the actual amount of material used during performance of a Process (see 4.2.86) .

4.2.131 Surface_roughness_requirement

A Surface_roughness_requirement is a type of Item_requirement (see 4.2.67) that specifies the degree of smoothness needed on the outer boundary of a part. The data associated with a Surface_roughness_requirement are the following:

- amount;
- measuring_direction.

4.2.131.1 amount

The amount specifies the quantitative measure of surface roughness that is allowable.

4.2.131.2 measuring_direction

The measuring_direction specifies the direction on the surface in which the roughness is to be measured. The measuring_direction need not be specified for a particular Surface_roughness_requirement. There may be more than one measuring_direction for a Surface_roughness_requirement.

4.2.132 Tensor_measure

A Tensor_measure is a quantitative measure specifying position within more than one coordinate system. The data associated with a Tensor_measure are the following:

- element.

The element specifies the component of the Tensor_measure. There may be more than one measuring_direction for a Tensor_measure.

4.2.133 Tolerance

A Tolerance is the amount of allowable deviation from an ideal value or configuration. Each Tolerance is either a Dimensional_tolerance (see 4.2.38), a Geometric_tolerance (see 4.2.46), or a Mismatch_tolerance (see 4.2.76). The data associated with a Tolerance are the following:

- name;
- specify_allowance_on.

4.2.133.1 name

The name specifies the label used to identify the Tolerance.

4.2.133.2 specify_allowance_on

The specify_allowance_on specifies the Casting_design_feature (see 4.2.9) to which the Tolerance applies. The specify_allowance_on need not be specified for a particular Tolerance. There may be more than one specify_allowance_on for a Tolerance.

4.2.134 Unit_process

An Unit_process is an operation used to produce a casting that is not usefully split into smaller operations.

4.2.135 Update_relationship

An Update_relationship is keeps track of the design modifications performed on an Item (see 4.2.66) The data associated with an Update_relationship are the following:

- original;
- updated.

4.2.135.1 original

The original specifies the initial design of an Item (see 4.2.66)

4.2.135.2 updated

The updated specifies the modified design of an Item (see 4.2.66)

4.2.136 Vector_measure

A Vector_measure is the value of a variable that is characterized by a magnitude and a direction. The data associated with a Vector_measure are the following:

- element.

The element specifies the component of the Vector_measure. There may be more than one updated for a Vector_measure.

4.2.137 Version

A Version is a particular form or variation of the design of an artifact or a process. Each Version is either an Item_version (see 4.2.68), or a Process_plan_version (see 4.2.91). The data associated with a Version are the following:

- description;
- identification.

4.2.137.1 description

The description specifies the written textual account of the Version.

4.2.137.2 identification

The identification specifies the label that uniquely identifies the Version.

4.3 Application assertions

This subclause specifies the application assertions for the Application Protocol Title application protocol. Application assertions specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects and UoFs. The application assertions and their definitions are given below.

4.3.1 Annotation to Document

Each Annotation has details described in zero, one, or many Document objects. Each Document describe details for zero, one, or many Annotation objects.

4.3.2 Approval to Date_and_time

Each Approval has the approval time specified by exactly one Date_and_time object. Each Date_and_time specifies the approval time for zero, one, or many Approval objects.

4.3.3 Approval to Person_and_organization

Each Approval is approved by exactly one Person_and_organization object. Each Person_and_organization approves zero, one, or many Approval objects.

4.3.4 Approval to Version

Each Approval approves exactly one Version object. Each Version is approved by zero, one, or many Approval objects.

4.3.5 Artifact to Date_and_time

Each Artifact has production date given by exactly one Date_and_time object. Each Date_and_time gives the production date for zero, one, or many Artifact objects.

4.3.6 Artifact to Heat

Each Artifact belongs to zero or one Heat objects. Each Heat contains zero, one, or many Artifact objects.

4.3.7 Artifact to Item_definition

Each Artifact is created using product specification given by exactly one Item_definition object. Each Item_definition gives the product specification to create zero, one, or many Artifact objects.

4.3.8 Artifact to Lot

Each Artifact belongs to zero or one Lot objects. Each Lot contains zero, one, or many Artifact objects.

4.3.9 Artifact to Process_plan

Each Artifact is produced using exactly one Process_plan object. Each Process_plan is used to produce zero, one, or many Artifact objects.

4.3.10 Assembly_item to Assembly_item_placement

Each Assembly_item has elements defined by one or many Assembly_item_placement objects. Each Assembly_item_placement defines the elements for zero, one, or many Assembly_item objects.

4.3.11 Assembly_item_placement to Axis_placement

Each Assembly_item_placement has placement location defined by exactly one Axis_placement object. Each Axis_placement defines the placement location for zero, one, or many Assembly_item_placement objects.

4.3.12 Assembly_item_placement to Item_definition

Each Assembly_item_placement has placed item defined by exactly one Item_definition object. Each Item_definition defines the placed item for zero, one, or many Assembly_item_placement objects.

4.3.13 Boundary_condition to Property_relationship

Each Boundary_condition is applied using exactly one Property_relationship object. Each Property_relationship is basis for applying zero, one, or many Boundary_condition objects.

4.3.14 Boundary_condition to Property_value

Each Boundary_condition is applied using exactly one Property_value object. Each Property_value is used to apply zero, one, or many Boundary_condition objects.

4.3.15 Boundary_condition to Shape_aspect

Each Boundary_condition is applied on exactly one Shape_aspect object. Each Shape_aspect has zero, one, or many Boundary_condition objects.

4.3.16 Casting_design_feature to Shape_aspect

Each Casting_design_feature has geometry modification outlined by exactly one Shape_aspect object. Each Shape_aspect outlines the geometry modification for zero, one, or many Casting_design_feature objects.

4.3.17 Casting_design_feature to Shape_representation

Each Casting_design_feature has resultant geometry described by exactly one Shape_representation object. Each Shape_representation describes the resultant geometry zero, one, or many Casting_design_feature objects.

4.3.18 Casting_machining_allowance_along_normal to Feature_parameter

Each Casting_machining_allowance_along_normal has thickness specified by exactly one Feature_parameter object. Each Feature_parameter specifies the thickness for zero, one, or many Casting_machining_allowance_along_normal objects.

4.3.19 Casting_machining_allowance_along_vector to Feature_parameter

Each Casting_machining_allowance_along_vector has offset amount specified by exactly one Feature_parameter object. Each Feature_parameter specifies the offset amount for zero, one, or many Casting_machining_allowance_along_vector objects.

4.3.20 Casting_machining_allowance_along_vector to Vector

Each Casting_machining_allowance_along_vector is represented by exactly one Vector object. Each Vector represents zero, one, or many Casting_machining_allowance_along_vector objects.

4.3.21 Casting_machining_with_explicit_surface to Shape_aspect

Each Casting_machining_with_explicit_surface has machining envelope specified by exactly one Shape_aspect object. Each Shape_aspect specifies the machining envelope for zero, one, or many Casting_machining_with_explicit_surface objects.

4.3.22 Casting_round_corner_transition to Feature_parameter

Each Casting_round_corner_transition has roundness value specified by exactly one Feature_parameter object. Each Feature_parameter specifies the roundness value for zero, one, or many Casting_round_corner_transition objects.

4.3.23 Casting_round_edge_transition to Feature_parameter

Each Casting_round_edge_transition has roundness value specified by exactly one Feature_parameter object. Each Feature_parameter specifies the roundness value for zero, one, or many Casting_round_edge_transition objects.

4.3.24 Casting_taper to Feature_parameter

Each Casting_taper has angle specified by zero or one Feature_parameter objects. Each Feature_parameter specifies the angle for zero, one, or many Casting_taper objects.

Each Casting_taper has ratio defined by zero or one Feature_parameter objects. Each Feature_parameter defines the ratio for zero, one, or many Casting_taper objects.

4.3.25 Casting_taper to Parting_surface

Each Casting_taper is placed with respect to exactly one Parting_surface object. Each Parting_surface specifies the placement of zero, one, or many Casting_taper objects.

4.3.26 Casting_taper to Shape_aspect

Each Casting_taper has edge reference described by exactly one Shape_aspect object. Each Shape_aspect describes the edge reference for zero, one, or many Casting_taper objects.

4.3.27 Composition_requirement to Substance

Each Composition_requirement contains zero or one Substance objects. Each Substance is a part of zero, one, or many Composition_requirement objects.

4.3.28 Compound_process to Process

Each Compound_process contains as subprocesses one or many Process objects. Each Process is a subprocess of zero, one, or many Compound_process objects.

4.3.29 Condition_or_assumption to Document

Each Condition_or_assumption is detailed in zero, one, or many Document objects. Each Document has details of zero, one, or many Condition_or_assumption objects.

4.3.30 Condition_or_assumption to Property_relationship

Each Condition_or_assumption is composed of zero, one, or many Property_relationship objects. Each Property_relationship is a portion of zero, one, or many Condition_or_assumption objects.

4.3.31 Condition_or_assumption to Property_value

Each Condition_or_assumption is composed of zero, one, or many Property_value objects. Each Property_value is a portion of zero, one, or many Condition_or_assumption objects.

4.3.32 Core to Core_master

Each Core is a portion of exactly one Core_master object. Each Core_master is composed of zero, one, or many Core objects.

4.3.33 Data_curve to Curve_point

Each Data_curve has data points represented by one or many Curve_point objects. Each Curve_point represents the data points for zero, one, or many Data_curve objects.

4.3.34 Data_curve to Unit

Each Data_curve has x-axis unit defined by exactly one Unit object. Each Unit defines the x-axis unit for zero, one, or many Data_curve objects.

Each Data_curve has y-axis unit defined by exactly one Unit object. Each Unit defines the y-axis unit for zero, one, or many Data_curve objects.

4.3.35 Defect_prediction to Axis_placement

Each Defect_prediction is located using zero or one Axis_placement objects. Each Axis_placement is used to locate zero, one, or many Defect_prediction objects.

4.3.36 Defect_prediction to Shape_aspect

Each Defect_prediction predict defects for zero or one Shape_aspect objects. Each Shape_aspect has its defects predicted by zero, one, or many Defect_prediction objects.

4.3.37 Design_request to Date_and_time

Each Design_request has the request date given by exactly one Date_and_time object. Each Date_and_time gives the request date for zero, one, or many Design_request objects.

4.3.38 Design_request to Item_definition

Each Design_request contains item defined by one or many Item_definition objects. Each Item_definition defines the item for zero, one, or many Design_request objects.

4.3.39 Design_request to Person_and_organization

Each Design_request is requested by exactly one Person_and_organization object. Each Person_and_organization requests zero, one, or many Design_request objects.

4.3.40 Design_update to Date_and_time

Each Design_update has the update date given by exactly one Date_and_time object. Each Date_and_time gives the update date for zero, one, or many Design_update objects.

4.3.41 Design_update to Design_request

Each Design_update is requested by one or many Design_request objects. Each Design_request requests zero, one, or many Design_update objects.

4.3.42 Design_update to Item_definition

Each Design_update is performed on exactly one Item_definition object. Each Item_definition is updated by zero, one, or many Design_update objects.

4.3.43 Design_update to Person_and_organization

Each Design_update is performed by exactly one Person_and_organization object. Each Person_and_organization performs zero, one, or many Design_update objects.

4.3.44 Die_mold_assembly to Shape_representation

Each Die_mold_assembly has the cavity shape represented by zero, one, or many Shape_representation objects. Each Shape_representation represents the cavity shape for zero, one, or many Die_mold_assembly objects.

4.3.45 Equipment_usage to Computer_file

Each Equipment_usage has instructions in zero or one Computer_file objects. Each Computer_file contains instructions for zero, one, or many Equipment_usage objects.

4.3.46 Equipment_usage to Machine_setting

Each Equipment_usage has zero, one, or many Machine_setting objects. Each Machine_setting is used by zero, one, or many Equipment_usage objects.

4.3.47 Failure_report to Artifact

Each Failure_report is composed of zero, one, or many Artifact objects. Each Artifact has zero, one, or many Failure_report objects.

4.3.48 Failure_report to Inspection_or_test_result

Each Failure_report describes exactly one Inspection_or_test_result object. Each Inspection_or_test_result is described in zero, one, or many Failure_report objects.

4.3.49 Feature_parameter to Measure_range_with_unit

Each Feature_parameter value is defined by exactly one Measure_range_with_unit object. Each Measure_range_with_unit defines the value for zero, one, or many Feature_parameter objects.

4.3.50 Flask to Measure_with_unit

Each Flask has length specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies the length for zero, one, or many Flask objects.

Each Flask has width specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies the width for zero, one, or many Flask objects.

Each Flask has cope height specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies cope height for zero, one, or many Flask objects.

Each Flask has drag height specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies drag height for zero, one, or many Flask objects.

4.3.51 Geometric_tolerance to Datum_reference_frame

Each Geometric_tolerance is specified with respect to zero or one Datum_reference_frame objects. Each Datum_reference_frame is used to specify zero, one, or many Geometric_tolerance objects.

4.3.52 Geometry_characterization to Axis_placement

Each Geometry_characterization has its orientation defined by zero or one Axis_placement objects. Each Axis_placement defines the orientation for zero, one, or many Geometry_characterization objects.

4.3.53 Geometry_characterization to Measure_with_unit

Each Geometry_characterization has size characterized by zero or one Measure_with_unit objects. Each Measure_with_unit characterizes the size for zero, one, or many Geometry_characterization objects.

4.3.54 Geometry_characterization to Shape_representation

Each Geometry_characterization has the shape defined by zero or one Shape_representation objects. Each Shape_representation defines the shape for zero, one, or many Geometry_characterization objects.

4.3.55 Geometry_requirement to Shape_representation

Each Geometry_requirement has the shape defined by exactly one Shape_representation object. Each Shape_representation defines the shape for zero, one, or many Geometry_requirement objects.

4.3.56 Heat to Date_and_time

Each Heat has production date given by exactly one Date_and_time object. Each Date_and_time gives the production date for zero, one, or many Heat objects.

4.3.57 Heat_treat_requirement to Data_curve

Each Heat_treat_requirement has data specified by zero or one Data_curve objects. Each Data_curve specifies the data for zero, one, or many Heat_treat_requirement objects.

4.3.58 Identification_marking to Graphics

Each Identification_marking is specified by exactly one Graphics object. Each Graphics specifies zero, one, or many Identification_marking objects.

4.3.59 Identification_marking to Measure_with_unit

Each Identification_marking height is defined by exactly one Measure_with_unit object. Each Measure_with_unit defines the height for zero, one, or many Identification_marking objects.

4.3.60 Inspection_or_test_requirement to Item_requirement

Each Inspection_or_test_requirement checks compliance with exactly one Item_requirement object. Each Item_requirement has compliance checked by zero, one, or many Inspection_or_test_requirement objects.

4.3.61 Inspection_or_test_requirement to Measure_with_unit

Each Inspection_or_test_requirement has frequency defined by exactly one Measure_with_unit object. Each Measure_with_unit defines frequency for zero, one, or many Inspection_or_test_requirement objects.

4.3.62 Inspection_or_test_result to Item_requirement

Each Inspection_or_test_result has results for zero, one, or many Item_requirement objects. Each Item_requirement has test results in zero, one, or many Inspection_or_test_result objects.

4.3.63 Inspection_or_test_result to Measure_with_unit

Each Inspection_or_test_result contains result defined by zero or one Measure_with_unit objects. Each Measure_with_unit defines results for zero, one, or many Inspection_or_test_result objects.

4.3.64 Inspection_or_test_result to Sampled_set

Each Inspection_or_test_result belongs to exactly one Sampled_set object. Each Sampled_set has one or many Inspection_or_test_result objects.

4.3.65 Inspection_or_test_result to Shape_aspect

Each Inspection_or_test_result applies to zero, one, or many Shape_aspect objects. Each Shape_aspect is checked by zero, one, or many Inspection_or_test_result objects.

4.3.66 Inspection_or_test_result to Statistical_measure

Each Inspection_or_test_result contains result defined by zero or one Statistical_measure objects. Each Statistical_measure defines results for zero, one, or many Inspection_or_test_result objects.

4.3.67 Inspection_or_test_result to Substance

Each Inspection_or_test_result contains results for zero or one Substance objects. Each Substance has zero, one, or many Inspection_or_test_result objects.

4.3.68 Integration_interval to Measure_with_unit

Each Integration_interval has time increment specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies the time increment for zero, one, or many Integration_interval objects.

Each Integration_interval has elapsed time interval defined by exactly one Measure_with_unit object. Each Measure_with_unit defines the elapsed time interval for zero, one, or many Integration_interval objects.

4.3.69 Investment_mold to Shape_representation

Each Investment_mold has the cavity shape represented by zero or one Shape_representation objects. Each Shape_representation represents the cavity shape for zero, one, or many Investment_mold objects.

4.3.70 Investment_pattern to Investment_pattern_die

Each Investment_pattern is defined by exactly one Investment_pattern_die object. Each Investment_pattern_die defines zero, one, or many Investment_pattern objects.

4.3.71 Item_definition to Item_requirement

Each Item_definition has requirements specified by one or many Item_requirement objects. Each Item_requirement specifies the requirements for zero, one, or many Item_definition objects.

4.3.72 Item_definition to Item_version

Each Item_definition is associated with zero or one Item_version objects. Each Item_version has zero, one, or many Item_definition objects.

4.3.73 Item_definition to Person_and_organization

Each Item_definition is created by zero or one Person_and_organization objects. Each Person_and_organization creates zero, one, or many Item_definition objects.

4.3.74 Item_requirement to Picture

Each Item_requirement is illustrated by zero, one, or many Picture objects. Each Picture illustrates zero, one, or many Item_requirement objects.

4.3.75 Item_requirement to Shape_aspect

Each Item_requirement is constrained by one or many Shape_aspect objects. Each Shape_aspect applies to zero, one, or many Item_requirement objects.

4.3.76 Item_version to Item

Each Item_version contains exactly one Item object. Each Item is a part of zero, one, or many Item_version objects.

4.3.77 Lot to Date_and_time

Each Lot has production date gevn by exactly one Date_and_time object. Each Date_and_time gives the production date of zero, one, or many Lot objects.

4.3.78 Machine_setting to Machine_setting_record

Each Machine_setting has setpoints described by exactly one Machine_setting_record object. Each Machine_setting_record describes the setpoints for zero, one, or many Machine_setting objects.

4.3.79 Machine_setting to Measure_with_unit

Each Machine_setting has parameters specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies parameters for zero, one, or many Machine_setting objects.

4.3.80 Machine_setting_record to Equipment_usage

Each Machine_setting_record records the context of exactly one Equipment_usage object. Each Equipment_usage has context recorded by zero, one, or many Machine_setting_record objects.

4.3.81 Machine_setting_record to Measure_with_unit

Each Machine_setting_record records the actual setting using exactly one Measure_with_unit object. Each Measure_with_unit is used to record actual settings in zero, one, or many Machine_setting_record objects.

4.3.82 Measure_range_with_unit to Measure_with_unit

Each Measure_range_with_unit has nominal amount specified by zero or one Measure_with_unit objects. Each Measure_with_unit specifies the nominal amount for zero, one, or many Measure_range_with_unit objects.

Each Measure_range_with_unit has lower limit specfied by zero or one Measure_with_unit objects. Each Measure_with_unit specifies the lower limit for zero, one, or many Measure_range_with_unit objects.

Each Measure_range_with_unit has upper limit specified by zero or one Measure_with_unit objects. Each Measure_with_unit specifies the upper limit for zero, one, or many Measure_range_with_unit objects.

4.3.83 Meshing_condition to Measure_range_with_unit

Each Meshing_condition has node density defined by zero or one Measure_range_with_unit objects. Each Measure_range_with_unit defines the node density for zero, one, or many Meshing_condition objects.

Each Meshing_condition has element density defined by zero or one Measure_range_with_unit objects. Each Measure_range_with_unit defines the element density for zero, one, or many Meshing_condition objects.

4.3.84 Meshing_condition to Shape_aspect

Each Meshing_condition has meshing area identified by zero, one, or many Shape_aspect objects. Each Shape_aspect identifies the meshing area for zero, one, or many Meshing_condition objects.

4.3.85 Mismatch_tolerance to Measure_with_unit

Each Mismatch_tolerance has maximum limit specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies the maximum limit for zero, one, or many Mismatch_tolerance objects.

4.3.86 Mismatch_tolerance to Parting_surface

Each Mismatch_tolerance is defined with respect to zero or one Parting_surface objects. Each Parting_surface is used to define zero, one, or many Mismatch_tolerance objects.

4.3.87 Parting_surface to Shape_aspect

Each Parting_surface has surface geometry specified by exactly one Shape_aspect object. Each Shape_aspect specifies the surface geometry for zero, one, or many Parting_surface objects.

4.3.88 Pattern to Pattern_master

Each Pattern is created using exactly one Pattern_master object. Each Pattern_master is used to create zero, one, or many Pattern objects.

4.3.89 Process to Equipment_usage

Each Process uses equipment specified by one or many Equipment_usage objects. Each Equipment_usage specifies the equipment for zero, one, or many Process objects.

4.3.90 Process to Item_definition

Each Process has output specified by zero, one, or many Item_definition objects. Each Item_definition specifies the output for zero, one, or many Process objects.

Each Process has input specifed by zero or one Item_definition objects. Each Item_definition specifies the input for zero, one, or many Process objects.

4.3.91 Process to Picture

Each Process is illustrated by zero, one, or many Picture objects. Each Picture illustrates zero, one, or many Process objects.

4.3.92 Process to Property_relationship

Each Process is composed of parameters related by zero, one, or many Property_relationship objects. Each Property_relationship specifies the interrelation of zero, one, or many Process objects.

4.3.93 Process to Property_value

Each Process has process parameter value defined by zero, one, or many Property_value objects. Each Property_value defines the process parameter value for zero, one, or many Process objects.

4.3.94 Process to Specification_reference

Each Process references zero, one, or many Specification_reference objects. Each Specification_reference is referenced by zero, one, or many Process objects.

4.3.95 Process to Substance_usage

Each Process has as input zero or one Substance_usage objects. Each Substance_usage is the input of zero, one, or many Process objects.

4.3.96 Process_execution to Process

Each Process_execution has process deviation defined by exactly one Process object. Each Process defines the deviated process for zero, one, or many Process_execution objects.

4.3.97 Process_execution to Process_plan

Each Process_execution uses plan outlined in exactly one Process_plan object. Each Process_plan outlines the plan for zero, one, or many Process_execution objects.

4.3.98 Process_execution to Sampled_set

Each Process_execution has the measured parts specified by zero or one Sampled_set objects. Each Sampled_set specifies the measured parts for zero, one, or many Process_execution objects.

4.3.99 Process_execution_record to Date_and_time

Each Process_execution_record has the recording date specified by zero or one Date_and_time objects. Each Date_and_time specifies the recording date for zero, one, or many Process_execution_record objects.

4.3.100 Process_execution_record to Process

Each Process_execution_record records the execution of exactly one Process object. Each Process has execution recorded by zero, one, or many Process_execution_record objects.

4.3.101 Process_parameter_record to Property_relationship

Each Process_parameter_record records the actual parameter of exactly one Property_relationship object. Each Property_relationship has its actual parameter recorded in zero, one, or many Process_parameter_record objects.

Each Process_parameter_record records the planned parameter of exactly one Property_relationship object. Each Property_relationship has its planned parameter recorded in zero, one, or many Process_parameter_record objects.

4.3.102 Process_parameter_record to Property_value

Each Process_parameter_record records the actual value of exactly one Property_value object. Each Property_value has its actual value recorded in zero, one, or many Process_parameter_record objects.

Each Process_parameter_record records the planned value of exactly one Property_value object. Each Property_value has its planned value recorded in zero, one, or many Process_parameter_record objects.

4.3.103 Process_plan to Date_and_time

Each Process_plan has creation time given by exactly one Date_and_time object. Each Date_and_time gives the creation time for zero, one, or many Process_plan objects.

4.3.104 Process_plan to Person_and_organization

Each Process_plan is created by zero or one Person_and_organization objects. Each Person_and_organization creates zero, one, or many Process_plan objects.

4.3.105 Process_plan to Process

Each Process_plan outlines the operation steps for one or many Process objects. Each Process is operated using steps outlined in zero, one, or many Process_plan objects.

4.3.106 Process_plan_version to Process_plan

Each Process_plan_version contains exactly one Process_plan object. Each Process_plan is part of zero, one, or many Process_plan_version objects.

4.3.107 Property_relationship to Data_curve

Each Property_relationship is represented by exactly one Data_curve object. Each Data_curve represents zero, one, or many Property_relationship objects.

4.3.108 Property_requirement to Property_relationship

Each Property_requirement has its property relationship defined by exactly one Property_relationship object. Each Property_relationship defines the property relationship for zero, one, or many Property_requirement objects.

4.3.109 Property_requirement to Property_value

Each Property_requirement has its property value defined by exactly one Property_value object. Each Property_value defines the property value for zero, one, or many Property_requirement objects.

4.3.110 Reporting_requirement to Item_requirement

Each Reporting_requirement is based on exactly one Item_requirement object. Each Item_requirement is the basis for zero, one, or many Reporting_requirement objects.

4.3.111 Request_for_change to Item_definition

Each Request_for_change suggests a change for zero or one Item_definition objects. Each Item_definition has change suggested by zero, one, or many Request_for_change objects.

4.3.112 Required_machining_allowance to Measure_with_unit

Each Required_machining_allowance value is specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies the value for zero, one, or many Required_machining_allowance objects.

4.3.113 Sampled_set to Artifact

Each Sampled_set is composed of inspected artifacts defined by zero, one, or many Artifact objects. Each Artifact defines inspected artifacts for zero, one, or many Sampled_set objects.

4.3.114 Sampled_set to Heat

Each Sampled_set has artifacts sampled from zero or one Heat objects. Each Heat contains artifacts used in zero, one, or many Sampled_set objects.

4.3.115 Sampled_set to Lot

Each Sampled_set has artifacts sampled from zero or one Lot objects. Each Lot contains artifacts used in zero, one, or many Sampled_set objects.

4.3.116 Sand_mold_component to Flask

Each Sand_mold_component is made using exactly one Flask object. Each Flask is used to make zero, one, or many Sand_mold_component objects.

4.3.117 Sand_mold_component to Pattern_and_rigging

Each Sand_mold_component has the pattern and rigging details outlined by exactly one Pattern_and_rigging object. Each Pattern_and_rigging outlines the pattern and rigging details for zero, one, or many Sand_mold_component objects.

4.3.118 Sand_mold to Shape_representation

Each Sand_mold has the cavity shape represented by zero, one, or many Shape_representation objects. Each Shape_representation represents the cavity shape for zero, one, or many Sand_mold objects.

4.3.119 Scalar_measure to Measure_with_unit

Each Scalar_measure has value specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies the value for zero, one, or many Scalar_measure objects.

4.3.120 Shape_aspect to Annotation

Each Shape_aspect is described by exactly one Annotation object. Each Annotation describes zero, one, or many Shape_aspect objects.

4.3.121 Simulated_property to Scalar_measure

Each Simulated_property has value specified by zero or one Scalar_measure objects. Each Scalar_measure specifies the value for zero, one, or many Simulated_property objects.

4.3.122 Simulated_property to Tensor_measure

Each Simulated_property has value specified by zero or one Tensor_measure objects. Each Tensor_measure specifies the value for zero, one, or many Simulated_property objects.

4.3.123 Simulated_property to Vector_measure

Each Simulated_property has value specified by zero or one Vector_measure objects. Each Vector_measure specifies the value for zero, one, or many Simulated_property objects.

4.3.124 Simulation_input to Computer_file

Each Simulation_input is present in zero or one Computer_file objects. Each Computer_file contains input data for zero, one, or many Simulation_input objects.

4.3.125 Simulation_input to Condition_or_assumption

Each Simulation_input is composed of zero, one, or many Condition_or_assumption objects. Each Condition_or_assumption is portion of zero, one, or many Simulation_input objects.

4.3.126 Simulation_input to Integration_interval

Each Simulation_input has integration parameter specified by zero, one, or many Integration_interval objects. Each Integration_interval specifies the integration parameter for zero, one, or many Simulation_input objects.

4.3.127 Simulation_input to Property_relationship

Each Simulation_input is composed of one or many Property_relationship objects. Each Property_relationship is a portion of zero, one, or many Simulation_input objects.

4.3.128 Simulation_input to Property_value

Each Simulation_input is composed of one or many Property_value objects. Each Property_value is a portion of zero, one, or many Simulation_input objects.

4.3.129 Simulation_input_region to Boundary_condition

Each Simulation_input_region is constrained by zero, one, or many Boundary_condition objects. Each Boundary_condition applies to zero, one, or many Simulation_input_region objects.

4.3.130 Simulation_input_region to Item_definition

Each Simulation_input_region has sections represented by zero, one, or many Item_definition objects. Each Item_definition represent sections for zero, one, or many Simulation_input_region objects.

4.3.131 Simulation_input_region to Meshing_condition

Each Simulation_input_region is composed of zero, one, or many Meshing_condition objects. Each Meshing_condition is a portion of zero, one, or many Simulation_input_region objects.

4.3.132 Simulation_input_region to Property_relationship

Each Simulation_input_region is composed of zero, one, or many Property_relationship objects. Each Property_relationship is a portion of zero, one, or many Simulation_input_region objects.

4.3.133 Simulation_input_region to Property_value

Each Simulation_input_region is composed of zero, one, or many Property_value objects. Each Property_value is a portion of zero, one, or many Simulation_input_region objects.

4.3.134 Simulation_input_region to Shape_representation

Each Simulation_input_region has geometry represented by exactly one Shape_representation object. Each Shape_representation represents the geometry for zero, one, or many Simulation_input_region objects.

4.3.135 Simulation_output to Defect_prediction

Each Simulation_output creates predicted values for zero, one, or many Defect_prediction objects. Each Defect_prediction is created by zero, one, or many Simulation_output objects.

4.3.136 Simulation_output to Simulation_result

Each Simulation_output creates simulation result in zero, one, or many Simulation_result objects. Each Simulation_result has the simulation result for zero, one, or many Simulation_output objects.

4.3.137 Simulation_output_region to Shape_representation

Each Simulation_output_region has shape specified by exactly one Shape_representation object. Each Shape_representation specifies the shape for zero, one, or many Simulation_output_region objects.

4.3.138 Simulation_output_region to Simulation_input_region

Each Simulation_output_region belongs to exactly one Simulation_input_region object. Each Simulation_input_region creates zero, one, or many Simulation_output_region objects.

4.3.139 Simulation_output_region to Simulation_unit_state

Each Simulation_output_region has simulation history defined by one or many Simulation_unit_state objects. Each Simulation_unit_state defines the simulation history for zero, one, or many Simulation_output_region objects.

4.3.140 Simulation_result to Measure_with_unit

Each Simulation_result begining time is defined by exactly one Measure_with_unit object. Each Measure_with_unit defines the beginning time for zero, one, or many Simulation_result objects.

Each Simulation_result ending time is defined by exactly one Measure_with_unit object. Each Measure_with_unit defines the ending time for zero, one, or many Simulation_result objects.

4.3.141 Simulation_result to Picture

Each Simulation_result displays the results using zero, one, or many Picture objects. Each Picture is used to display results for zero, one, or many Simulation_result objects.

4.3.142 Simulation_result to Simulation_output_region

Each Simulation_result is composed of one or many Simulation_output_region objects. Each Simulation_output_region is a portion of zero, one, or many Simulation_result objects.

4.3.143 Simulation_run to Process

Each Simulation_run has associated process identified by zero or one Process objects. Each Process is associated with zero, one, or many Simulation_run objects.

4.3.144 Simulation_run to Simulation_input

Each Simulation_run has input represented by exactly one Simulation_input object. Each Simulation_input represents the input for zero, one, or many Simulation_run objects.

4.3.145 Simulation_run to Simulation_input_region

Each Simulation_run is composed of one or many Simulation_input_region objects. Each Simulation_input_region is a portion of zero, one, or many Simulation_run objects.

4.3.146 Simulation_run to Simulation_output

Each Simulation_run creates exactly one Simulation_output object. Each Simulation_output is created by zero, one, or many Simulation_run objects.

4.3.147 Simulation_run to Simulation_run_relationship

Each Simulation_run has relationship represented by exactly one Simulation_run_relationship object. Each Simulation_run_relationship represents the relationship for zero, one, or many Simulation_run objects.

4.3.148 Simulation_run to Simulation_software

Each Simulation_run uses software identified by exactly one Simulation_software object. Each Simulation_software identifies the software for zero, one, or many Simulation_run objects.

4.3.149 Simulation_unit to Axis_placement

Each Simulation_unit geometry is located using exactly one Axis_placement object. Each Axis_placement is used to locate the geometry for zero, one, or many Simulation_unit objects.

4.3.150 Simulation_unit to Shape_aspect

Each Simulation_unit has the unit shape described by zero or one Shape_aspect objects. Each Shape_aspect describes the unit shape for zero, one, or many Simulation_unit objects.

4.3.151 Simulation_unit to Simulation_output_region

Each Simulation_unit belongs to exactly one Simulation_output_region object. Each Simulation_output_region is owned by zero, one, or many Simulation_unit objects.

4.3.152 Simulation_unit to Simulation_unit_state

Each Simulation_unit has the unit history defined by one or many Simulation_unit_state objects. Each Simulation_unit_state defines the unit history for zero, one, or many Simulation_unit objects.

4.3.153 Simulation_unit_state to Measure_with_unit

Each Simulation_unit_state has evaluation time specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies the evaluation time for zero, one, or many Simulation_unit_state objects.

4.3.154 Simulation_unit_state to Simulated_property

Each Simulation_unit_state is composed of one or many Simulated_property objects. Each Simulated_property is a portion of zero, one, or many Simulation_unit_state objects.

4.3.155 Specification to Organization

Each Specification is issued by exactly one Organization object. Each Organization issues zero, one, or many Specification objects.

4.3.156 Specification_reference to Specification

Each Specification_reference refers to exactly one Specification object. Each Specification is referenced by zero, one, or many Specification_reference objects.

Each Specification_reference refers to exactly one Specification object. Each Specification is referenced by zero, one, or many Specification_reference objects.

4.3.157 Statistical_measure to Measure_with_unit

Each Statistical_measure has mean value defined by exactly one Measure_with_unit object. Each Measure_with_unit defines the mean value for zero, one, or many Statistical_measure objects.

Each Statistical_measure has variance defined by exactly one Measure_with_unit object. Each Measure_with_unit defines the variance for zero, one, or many Statistical_measure objects.

4.3.158 Substance to Substance_composition_element

Each Substance is composed of zero, one, or many Substance_composition_element objects. Each Substance_composition_element belongs to one or many Substance objects.

4.3.159 Substance to Substance_structure

Each Substance is formed by zero or one Substance_structure objects. Each Substance_structure forms zero, one, or many Substance objects.

4.3.160 Substance_composition_element to Measure_with_unit

Each Substance_composition_element has composition amount defined by exactly one Measure_with_unit object. Each Measure_with_unit defines the composition amount for zero, one, or many Substance_composition_element objects.

4.3.161 Substance_composition_element_record to Measure_with_unit

Each Substance_composition_element_record uses the composition amount specified by exactly one Measure_with_unit object. Each Measure_with_unit specifies the composition amount to be used by zero, one, or many Substance_composition_element_record objects.

4.3.162 Substance_composition_element_record to Substance_usage

Each Substance_composition_element_record records the usage context of exactly one Substance_usage object. Each Substance_usage has usage context recorded by zero, one, or many Substance_composition_element_record objects.

4.3.163 Substance_structure to Picture

Each Substance_structure is illustrated by exactly one Picture object. Each Picture illustrates zero, one, or many Substance_structure objects.

4.3.164 Substance_structure to Specification_reference

Each Substance_structure is referenced by exactly one Specification_reference object. Each Specification_reference refers to zero, one, or many Substance_structure objects.

4.3.165 Substance_structure to Substance_structure_element

Each Substance_structure has elements defined by one or many Substance_structure_element objects. Each Substance_structure_element defines the elements for zero, one, or many Substance_structure objects.

4.3.166 Substance_structure_element to Geometry_characterization

Each Substance_structure_element has geometry represented by zero, one, or many Geometry_characterization objects. Each Geometry_characterization represents the geometry for zero, one, or many Substance_structure_element objects.

4.3.167 Substance_structure_element to Measure_with_unit

Each Substance_structure_element contains amount described by exactly one Measure_with_unit object. Each Measure_with_unit describes the amount for zero, one, or many Substance_-

structure_element objects.

4.3.168 Substance_structure_element to Substance

Each Substance_structure_element is a part of exactly one Substance object. Each Substance contains zero, one, or many Substance_structure_element objects.

4.3.169 Substance_usage to Measure_with_unit

Each Substance_usage has amount specified by zero or one Measure_with_unit objects. Each Measure_with_unit specifies the amount for zero, one, or many Substance_usage objects.

4.3.170 Substance_usage to Substance

Each Substance_usage specifies the usage for exactly one Substance object. Each Substance has usage specified by zero, one, or many Substance_usage objects.

4.3.171 Substance_usage_record to Measure_with_unit

Each Substance_usage_record records the actual usage amount using exactly one Measure_with_unit object. Each Measure_with_unit is used to record the actual usage amount in zero, one, or many Substance_usage_record objects.

4.3.172 Surface_roughness_requirement to Curve_on_surface

Each Surface_roughness_requirement has the measuring direction specified by zero, one, or many Curve_on_surface objects. Each Curve_on_surface specifies the measuring direction for zero, one, or many Surface_roughness_requirement objects.

4.3.173 Surface_roughness_requirement to Measure_range_with_unit

Each Surface_roughness_requirement amount is specified by exactly one Measure_range_with_unit object. Each Measure_range_with_unit specifies the amount for zero, one, or many Surface_roughness_requirement objects.

4.3.174 Tensor_measure to Measure_with_unit

Each Tensor_measure has between 1 and 9 Measure_with_unit objects. Each Measure_with_unit defines the elements for zero, one, or many Tensor_measure objects.

4.3.175 Tolerance to Casting_design_feature

Each Tolerance specifies allowance for zero, one, or many Casting_design_feature objects. Each Casting_design_feature allowance is specified by zero, one, or many Tolerance objects.

4.3.176 Update_relationship to Item_definition

Each Update_relationship stores the original design of exactly one Item_definition object. Each Item_definition has the original design stored in zero, one, or many Update_relationship objects.

Each Update_relationship stores the updated design of exactly one Item_definition object. Each Item_definition has the updated design stored in zero, one, or many Update_relationship objects.

4.3.177 Vector_measure to Measure_with_unit

Each Vector_measure has between 1 and 3 Measure_with_unit objects. Each Measure_with_unit defines the elements for zero, one, or many Vector_measure objects.

5 Application interpreted model

5.1 Mapping table

This clause contains the mapping table that shows how each UoF and application object of this part of ISO 10303 (see clause 4) maps to one or more AIM constructs (see annex A). The mapping table is organized in five columns.

- Column 1) Application element: Name of an application element as it appears in the application object definition in 4.2. Application object names are written in uppercase. Attribute names and assertions are listed after the application object to which they belong and are written in lower case.

- Column 2) AIM element: Name of an AIM element as it appears in the AIM (see annex A), the term 'IDENTICAL MAPPING', or the term 'PATH'. AIM entities are written in lower case. Attribute names of AIM entities are referred to as <entity name> . <attribute name>. The mapping of an application element may result in several related AIM elements. Each of these AIM elements requires a line of its own in the table. The term 'IDENTICAL MAPPING' indicates that both application objects of an application assertion map to the same AIM element. The term 'PATH' indicates that the application assertion maps to the entire reference path.

- Column 3) Source: For those AIM elements that are interpreted from the integrated resources, this is the number of the corresponding part of ISO 10303. For those AIM elements that are created for the purpose of this part of ISO 10303, this is the number of this part.

– Column 4) Rules: One or more numbers may be given that refer to rules that apply to the current AIM element or reference path. For rules that are derived from relationships between application objects, the same rule is referred to by the mapping entries of all the involved AIM elements. The expanded names of the rules are listed after the table.

– Column 5) Reference path: To describe fully the mapping of an ARM element, it may be necessary to specify a reference path through several related AIM elements. The reference path column documents the role of an AIM element relative to the AIM element in the row succeeding it. Two or more such related AIM elements define the interpretation of the integrated resources that satisfies the requirement specified by the application object. For each AIM element that has been created for use within this part of ISO 10303, a reference path up to its supertype from an integrated resource is specified.

For the expression of reference paths the following notational conventions apply:

a) `[]` : multiple AIM elements or sections of the reference path are required to satisfy an information requirement;

b) `()` : multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;

c) `{}` : enclosed section constrains the reference path to satisfy an information requirement;

d) `->` : attribute references the entity or select type given in the following row;

e) `<-` : entity or select type is referenced by the attribute in the following row;

f) `[i]` : attribute is an aggregation of which a single member is given in the following row;

g) `[n]` : attribute is an aggregation of which member `n` is given in the following row;

h) `=>` : entity is a supertype of the entity given in the following row;

i) `<=` : entity is a subtype of the entity given in the following row;

j) `=` : the string, select, or enumeration type is constrained to a choice or value;

k) `\` : the line continuations for strings that wrap.

Table 1 – Mapping table for feature UoF

Application element	AIM element	Source	Rules	Reference path
CASTING_DESIGN_ FEATURE	(casting_machining_allowance) (casting_round_corner_ transition) (casting_round_edge_transition) (casting_taper) (identification_marking)	223 223 223 223 223		(casting_machining_allowance <=) (casting_round_corner_transition <=) (casting_round_edge_transition <=) (casting_taper <=) (identification_marking <=) shape_aspect
casting_design_feature to shape_aspect (as geometry_to_be_modified)	PATH			(casting_machining_allowance <=) (casting_round_corner_transition <=) (casting_round_edge_transition <=) (casting_taper <=) (identification_marking <=) shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'geometry modification'} shape_aspect_relationship.relating_shape_aspect -> shape_aspect
casting_design_feature to shape_representation (as resultant_geometry)	PATH			(casting_machining_allowance <=) (casting_round_corner_transition <=) (casting_round_edge_transition <=) (casting_taper <=) (identification_marking <=) shape_aspect shape_definition = shape_aspect

Table 1 – Continued

Application element	AIM element	Source	Rules	Reference path
				shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'resultant geometry'} representation => shape_representation => (advanced_brep_shape_representation) (faceted_brep_shape_representation) (elementary_brep_shape_representation) (geometrically_bounded_2d_shape_representation)
CASTING_MACHINING_ ALLOWANCE	casting_machining_allowance	223		casting_machining_allowance <= shape_aspect {shape_aspect (shape_aspect.description = 'allowance along normal') (shape_aspect.description = 'allowance along vector') (shape_aspect.description = 'explicit surface allowance')
CASTING_MACHINING_ ALLOWANCE_ALONG_ NORMAL	casting_machining_allowance	223		casting_machining_allowance <= shape_aspect {shape_aspect shape_aspect.description = 'allowance along normal'}
casting_machining_allowance_ along_normal to feature_parameter (as thickness)	PATH			casting_machining_allowance <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation representation.name = 'feature parameters']

Table 1 – Continued

Application element	AIM element	Source	Rules	Reference path
				[representation representation.items[i] -> [representation_item representation_item.name = 'thickness'] [representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit]]}
CASTING_MACHINING_ ALLOWANCE_ALONG_ VECTOR	casting_machining_allowance	223		casting_machining_allowance <= shape_aspect {shape_aspect shape_aspect.description = 'allowance along vector'}
casting_machining_allowance_ along_vector to feature_parameter (as offset_amount)	PATH			casting_machining_allowance <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation representation.name = 'feature parameters'] [representation representation.items[i] -> [representation_item representation_item.name = 'offset amount'] [representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit]]}
casting_machining_allowance_ along_vector to vector (as direction)	PATH			casting_machining_allowance <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition

Table 1 – Continued

Application element	AIM element	Source	Rules	Reference path
				property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'feature parameters'} representation representation.items[i] -> representation_item => geometric_representation_item => vector
CASTING_MACHINING_ WITH_EXPLICIT_SURFACE	casting_machining_allowance	223		casting_machining_allowance <= shape_aspect {shape_aspect shape_aspect.description = 'explicit surface allowance'}
casting_machining_with_ explicit_surface to shape_aspect (as enveloping_geometry)	PATH			casting_machining_allowance <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'enveloping geometry'} shape_aspect_relationship.related_shape_aspect -> shape_aspect
CASTING_ROUND_ CORNER_TRANSITION	casting_round_corner_transition	223		casting_round_corner_transition <= shape_aspect
casting_round_corner_transition to feature_parameter (as radius)	PATH			casting_round_corner_transition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation representation.name = 'feature parameters']

Table 1 – Continued

Application element	AIM element	Source	Rules	Reference path
				[representation representation.items[i] -> [representation_item representation_item.name = 'radius'] [representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit]]}
CASTING_ROUND_EDGE_TRANSITION	casting_round_edge_transition	223		casting_round_edge_transition <= shape_aspect
casting_round_edge_transition to feature_parameter (as radius)	PATH			casting_round_edge_transition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation representation.name = 'feature parameters'] [representation representation.items[i] -> [representation_item representation_item.name = 'radius'] [representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit]]}
CASTING_TAPER	casting_taper	223		casting_taper <= shape_aspect
method	descriptive_representation_item.-	45		casting_taper <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition

Table 1 – Continued

Application element	AIM element	Source	Rules	Reference path
				property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'feature parameters'} representation representation.items[i] -> {representation_item representation_item.name = 'taper method'} representation_item => descriptive_representation_item descriptive_representation_item.description {(descriptive_representation_item.description = 'average') (descriptive_representation_item.description = 'minus') (descriptive_representation_item.description = 'plus')}
casting_taper to feature_parameter (as angle)	PATH			casting_taper <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation representation.name = 'feature parameters'] [representation representation.items[i] -> [representation_item representation_item.name = 'angle'] [representation_item => measure_representation_item <= measure_with_unit =>

Table 1 – Continued

Application element	AIM element	Source	Rules	Reference path
				plane_angle_measure_with_unit]]}
casting_taper to feature_parameter (as ratio)	PATH			casting_taper <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation representation.name = 'feature parameters'] [representation representation.items[i] -> [representation_item representation_item.name = 'ratio'] [representation_item => measure_representation_item <= measure_with_unit => ratio_measure_with_unit]]}
casting_taper to parting_surface (as with_respect_to)	PATH			casting_taper <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'parting surface'}
casting_taper to shape_aspect (as reference_edge)	PATH			casting_taper <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect

Table 1 – Continued

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.description = ‘reference edge’}
FEATURE_PARAMETER	representation	43		{representation representation.name = ‘feature parameters’}
feature_parameter to measure_range_with_unit (as value)	PATH			representation representation.items[i] -> {representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier (type_qualifier.name = ‘nominal amount’) (type_qualifier.name = ‘lower limit’) (type_qualifier.name = ‘upper limit’)} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
IDENTIFICATION_MARKING	identification_marking	223		identification_marking <= shape_aspect
raised	descriptive_representation_item.-	45		identification_marking <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = ‘raised’} representation_item => descriptive_representation_item descriptive_representation_item.description {(descriptive_representation_item.description = ‘raised’) (descriptive_representation_item.description = ‘not raised’)}

Table 1 – Continued

Application element	AIM element	Source	Rules	Reference path
identification_marking to graphics (as marking)	PATH			identification_marking document_assigned_item = identification_marking document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type.product_data_type = 'graphics'}
identification_marking to measure_with_unit (as height)	PATH			identification_marking <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'height'} representation_item => measure_representation_item <= {measure_with_unit => length_measure_with_unit} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
PARTING_SURFACE	shape_aspect	41		{shape_aspect shape_aspect.description = 'parting surface'}

Table 1 – Concluded

Application element	AIM element	Source	Rules	Reference path
parting_surface to shape_aspect (as surface_geometry)	IDENTICAL MAPPING			

Table 2 – Mapping table for geometry UoF

Application element	AIM element	Source	Rules	Reference path
ANNOTATION	representation	43		{representation representation.name = ‘shape aspect annotation’}
description	descriptive_representation_ item.-	45		representation representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
name	representation_item.name	43		representation representation.items[i] -> {representation_item => descriptive_representation_item} representation_item representation_item.name
annotation to document (as more_information)	PATH			representation document_assigned_item = representation document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document -> document
annotation to shape_aspect (as applies_to)	PATH		1	representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect
AXIS_PLACEMENT	placement	42		

Table 2 – Concluded

Application element	AIM element	Source	Rules	Reference path
B_REP_SHAPE_ REPRESENTATION	(advanced_brep_shape_ representation) (faceted_brep_shape_ representation)(elementary_ brep_shape_representation) (elementary_brep_shape_ representation)	514 512 513		(advanced_brep_shape_representation <=) (faceted_brep_shape_representation <=) (elementary_brep_shape_representation <=) shape_representation
CURVE_ON_SURFACE	surface_curve	42		
GEOMETRIC_2D_SHAPE_ REPRESENTATION	geometrically_bounded_2d_ shape_representation	503		geometrically_bounded_2d_shape_representation <= shape_representation
SHAPE_ASPECT	shape_aspect	41		
SHAPE_REPRESENTATION	(advanced_brep_shape_ representation) (faceted_brep_shape_ representation) (elementary_brep_shape_ representation) (geometrically_bounded_2d_ shape_representation)	514 512 513 503		(advanced_brep_shape_representation <=) (faceted_brep_shape_representation <=) (elementary_brep_shape_representation <=) (geometrically_bounded_2d_shape_representation <=) shape_representation
VECTOR	vector	42		

Table 3 – Mapping table for measurement UoF

Application element	AIM element	Source	Rules	Reference path
MEASURE_RANGE_WITH_UNIT	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41		
measure_range_with_unit to measure_with_unit (as lower_limit)	IDENTICAL MAPPING			{[measure_with_unit.value_component] [measure_with_unit.unit_component] (measure_with_unit <- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[i] ->) (measure_with_unit => measure_representation_item <= representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] ->) value_qualifier value_qualifier = type_qualifier type_qualifier type_qualifier.name = 'lower limit'}
measure_range_with_unit to measure_with_unit (as nominal_amount)	IDENTICAL MAPPING			{[measure_with_unit.value_component] [measure_with_unit.unit_component] (measure_with_unit <- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[i] ->) (measure_with_unit => measure_representation_item <= representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] ->) value_qualifier value_qualifier = type_qualifier type_qualifier

Table 3 – Continued

Application element	AIM element	Source	Rules	Reference path
				type_qualifier.name = ‘nominal amount’}
measure_range_with_unit to measure_with_unit (as upper_limit)	IDENTICAL MAPPING			{[measure_with_unit.value_component] [measure_with_unit.unit_component] (measure_with_unit <- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[i] ->) (measure_with_unit => measure_representation_item <= representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] ->) value_qualifier value_qualifier = type_qualifier type_qualifier type_qualifier.name = ‘upper limit’}
MEASURE_WITH_UNIT	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41		
SCALAR_MEASURE	scalar	104		
scalar_measure to measure_with_unit (as scalar)	IDENTICAL MAPPING			
STATISTICAL_MEASURE	statistical_measure	223		statistical_measure <= representation
number_of_measurements	measure_with_unit.value_component	41		statistical_measure <= representation representation.items[i] -> {representation_item representation_item.name = ‘number of measurements’} representation_item => measure_representation_item <=

Table 3 – Continued

Application element	AIM element	Source	Rules	Reference path
				measure_with_unit measure_with_unit.value_component {measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
statistical_measure to measure_with_unit (as mean)	PATH			statistical_measure <= representation representation.items[i] -> {representation_item representation_item.name = 'mean'} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
statistical_measure to measure_with_unit (as variance)	PATH			statistical_measure <= representation representation.items[i] -> {representation_item representation_item.name = 'variance'} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
TENSOR_MEASURE	(symmetric_tensor2_2d)	104		
	(symmetric_tensor2_3d)	104		
	(anisotropic_symmetric_tensor2_2d)	104		
	(anisotropic_symmetric_tensor2_3d)	104		
	(isotropic_symmetric_tensor2_3d)	104		

Table 3 – Concluded

Application element	AIM element	Source	Rules	Reference path
	(orthotropic_symmetric_tensor2_3d)	104		
tensor_measure to measure_with_unit (as elements)	IDENTICAL MAPPING			
UNIT	named_unit			
VECTOR_MEASURE	tensor1	104		
vector_measure to measure_with_unit (as elements)	IDENTICAL MAPPING			

Table 4 – Mapping table for process_plan UoF

Application element	AIM element	Source	Rules	Reference path
COMPOUND_PROCESS	action_method	41		{(action_method <-) (action_method <- action_method_relationship.related_action_method action_method_relationship {action_method_relationship } action_method_relationship.relating_action_method -> action_method <-) action.chosen_method action => (product_definition_process) (product_property_process)}
compound_process to process (as elements)	PATH			action_method <- action_method_relationship.relating_action_method {[action_method_relationship action_method_relationship.name = 'process list'] [(action_method_relationship) (action_method_relationship => concurrent_action_method) (action_method_relationship => serial_action_method)] action_method_relationship action_method_relationship.related_action_method -> action_method
EQUIPMENT_USAGE	1: (action_resource_requirement) 2: (action_resource)	49 41		1: ({action_resource_requirement action_resource_requirement.kind -> type_or_capability type_or_capability = .TYPE_REQUIREMENT.}) 2: ({action_resource action_resource.kind -> action_resource_type

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
				action_resource_type.name = 'equipment'})
equipment_name	1: (action_resource_requirement.name) 2: (action_resource.name)	49 41		
identification	1: (action_resource_requirement.description) 2: (action_resource.description)	49 41		
operation_instruction	action_method.description	41		1: (action_resource_requirement action_resource_requirement.operation -> characterized_action_definition characterized_action_definition = action_method) 2: (action_resource action_resource.usage[i] -> supported_item supported_item = action_method) action_method action_method.description
equipment_usage to computer_file (as machine_program)	PATH			1: (action_resource_requirement document_assigned_item = action_resource_requirement) 2: (action_resource document_assigned_item = action_resource) document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type.product_data_type = 'computer file'}
equipment_usage to machine_setting (as equipment_settings)	PATH			1: (action_resource_requirement characterized_action_resource = action_resource_requirement) 2: (action_resource characterized_action_resource = action_resource) characterized_action_resource <- resource_property.resource resource_property <-

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
				resource_property_representation.property resource_property_representation resource_property_representation.representation -> representation {representation representation.name = ‘machine setting’}
MACHINE_SETTING	representation	43		{representation representation.name = ‘machine setting’}
parameter_name	representation_item.name	43		representation representation.items[i] -> representation_item representation_item.name
machine_setting to measure_with_unit (as parameter_setting)	PATH			representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
PROCESS	action_method			{(action_method <-) (action_method <- action_method_relationship.related_action_method action_method_relationship {action_method_relationship } {action_method_relationship => (concurrent_action_method) (serial_action_method) (serial_action_method => sequential_method)} action_method_relationship.relating_action_method -> action_method <-) action.chosen_method action => (product_definition_process) (product_property_process)}
description	action_method.purpose	41		

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
identification	NO MAPPING			
name	action_method.name	41		
process to equipment_usage (as equipment_used)	PATH			action_method 1: (characterized_action_definition = action_method characterized_action_definition <- action_resource_requirement.operation action_resource_requirement) 2: (supported_item = action_method supported_item <- action_resource.usage[i] action_resource)
process to item_definition (as input_items)	PATH			action_method <- action.chosen_method action <- action_assignment.assigned_action action_assignment => process_plan_input_assignment process_plan_input_assignment.items[i] -> process_plan_input_item (process_plan_input_item = characterized_object characterized_object) (process_plan_input_item = product_definition product_definition) (process_plan_input_item = product_definition product_definition => casting_part_definition) (process_plan_input_item = product_definition product_definition => casting_mold_design_element) (process_plan_input_item = product_definition product_definition => die_mold) (process_plan_input_item = product_definition product_definition => investment_mold) (process_plan_input_item = product_definition product_definition => investment_mold_pattern) (process_plan_input_item = product_definition

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition => investment_mold_sprue) (process_plan_input_item = product_definition product_definition => sand_mold)
process to item_definition (as output_items)	PATHsee note 2			(action_method <-) (action_method <- action_method_relationship.related_action_method action_method_relationship {action_method_relationship } action_method_relationship.relating_action_method -> action_method <-) action.chosen_method action => (product_definition_process <- process_product_association.process process_product_association process_product_association.defined_product -> characterized_product_definition characterized_product_definition = product_definition (product_definition) (product_definition => casting_part) (product_definition => casting_mold_design_element) (product_definition => die_mold) (product_definition => investment_mold) (product_definition => investment_mold_pattern) (product_definition => investment_mold_sprue) (product_definition => sand_mold)) (product_property_process <- process_property_association.process process_property_association process_property_association.property -> property_definition

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition (characterized_definition = characterized_object characterized_object) (characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition (product_definition) (product_definition => casting_part) (product_definition => casting_mold_design_element) (product_definition => die_mold) (product_definition => investment_mold) (product_definition => investment_mold_pattern) (product_definition => investment_mold_sprue) (product_definition => sand_mold)))
process to picture (as illustrations)	PATH			(action_method document_assigned_item = action_method) (action_method <- action_method_relationship.related_action_method action_method_relationship {action_method_relationship } action_method_relationship.relating_action_method -> action_method <- action.chosen_method action => (product_definition_process document_assigned_item = product_definition_process) (product_property_process document_assigned_item = product_property_process))

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
				document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type.product_data_type = 'picture'}
process to property_relationship (as process_parameters)	PATH			action_method characterized_action_definition = action_method characterized_action_definition <- action_property.definition {action_property action_property.name = 'process parameter'} action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> {representation representation.name = 'process parameter representation'} representation => data_curve_representation
process to property_value (as process_parameters)	PATH			action_method characterized_action_definition = action_method characterized_action_definition <- action_property.definition {action_property action_property.name = 'process parameter'} action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> {representation representation.name = 'process parameter representation'} representation representation.items[i] ->

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation_item => measure_representation_item
process to specification_reference (as reference_to)	PATH			action_method document_assigned_item = action_method document_assigned_item <- casting_document_reference.items[i] casting_document_reference {casting_document_reference <= document_reference document_reference.assigned_document -> document document.kind -> document_type document_type.product_data_type = 'specification'}
process to substance_usage (as input_items)	PATH			action_method <- action.chosen_method action <- action_assignment.assigned_action action_assignment => process_plan_input_assignment process_plan_input_assignment.items[i] -> process_plan_input_item process_plan_input_item = product_definition product_definition <- product_definition_relationship.related_product_- definition
PROCESS_PLAN	process_plan_version	223		process_plan_version <= action
description	action.description			process_plan_version <= action action.description
identification	NO MAPPING			
name	action.name			process_plan_version <= action action.name
process_plan to date_and_time (as designed_date)	PATH			process_plan_version 1: (date_assigned_item = process_plan_version date_assigned_item <- casting_date_assignment.items[i] casting_date_assignment <= date_assignment {date_assignment

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
				date_assignment.role -> date_role date_role.name = ‘design date’} date_assignment.assigned_date -> date => calendar_date) 2: (date_and_time_assigned_item = process_plan_version date_and_time_assigned_item <- casting_date_and_time_assignment.items[i] casting_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = ‘design date’} date_and_time_assignment.assigned_date_and_time -> date_and_time)
process_plan to person_and_organization (as designer)	PATH			process_plan_version person_and_organization_assigned_item = process_plan_version person_and_organization_assigned_item <- casting_person_and_organization_assignment.items[i] casting_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = ‘designer’} person_and_organization_assignment.assigned_person_- and_organization -> person_and_organization
process_plan to process (as operations)	PATH			process_plan_version <= action action.chosen_method -> (action_method) (action_method <- action_method_relationship.relating_action_method action_method_relationship

Table 4 – Continued

Application element	AIM element	Source	Rules	Reference path
				{action_method_relationship } {action_method_relationship => (concurrent_action_method) (serial_action_method) (serial_action_method => sequential_method)} action_method_relationship.related_action_method -> action_method)
SUBSTANCE_USAGE	product_definition_relationship.related_product_definition	41		{product_definition_relationship.related_product_- definition product_definition_relationship => product_definition_usage => make_from_usage_option}
substance_usage to measure_range_with_unit (as amount)	PATH			product_definition_relationship.related_product_- definition product_definition_relationship => product_definition_usage => make_from_usage_option make_from_usage_option.quantity -> {measure_with_unit <- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier (type_qualifier.name = 'nominal amount') (type_qualifier.name = 'lower limit') (type_qualifier.name = 'upper limit')} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
substance_usage to substance (as made_of)	PATH			product_definition_relationship.related_product_- definition -> product_definition {product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <-

Table 4 – Concluded

Application element	AIM element	Source	Rules	Reference path
				product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'material'}
UNIT_PROCESS	action_method	41		{(action_method <-) (action_method <- action_method_relationship.related_action_method action_method_relationship {action_method_relationship } action_method_relationship.relating_action_method -> action_method <-) action.chosen_method action => (product_definition_process) (product_property_process)}

Table 5 – Mapping table for product_management UoF

Application element	AIM element	Source	Rules	Reference path
APPROVAL	approval	41	5	
approval to date_and_time (as when_approved)	PATH		3	approval <- approval_date_time.dated_approval approval_date_time approval_date_time.date_time -> date_time_select 1: (date_time_select = date date => calendar_date) 2: (date_time_select = date_and_time date_and_time)
approval to person_and_organization (as approved_by)	PATH		4	approval <- approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization -> person_organization_select person_organization_select = person_and_organization person_and_organization
approval to version (as approved_version)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => casting_approval_assignment casting_approval_assignment.items[i] -> approval_assigned_item (approval_assigned_item = product_definition_formation product_definition_formation) (approval_assigned_item = process_plan_version process_plan_version)
COMPUTER_FILE	document	41		{document document.kind -> document_type document_type.product_data_type = 'computer file'}

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
file_name	document.id	41		
CURVE_POINT	cartesian_point	42		
x_value	cartesian_point.coordinates[1]	42		
y_value	cartesian_point.coordinates[2]	42		
DATA_CURVE	data_curve_representation	223		data_curve_representation <= representation
interpolation_method	descriptive_representation_item.description	45		data_curve_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'interpolation method'} representation_item => descriptive_representation_item descriptive_representation_item.description
data_curve to curve_point (as data_points)	PATH			data_curve_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => (curve => bounded_curve => polyline polyline.points[i] ->) (point =>) cartesian_point
data_curve to unit (as x_unit)	PATH			data_curve_representation <= representation representation.context_of_items -> representation_context => global_unit_assigned_context global_unit_assigned_context.units[1] -> unit unit = derived_unit derived_unit derived_unit.elements[1] -> derived_unit_element derived_unit_element.unit -> named_unit

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
data_curve to unit (as y_unit)	PATH			data_curve_representation <= representation representation.context_of_items -> representation_context => global_unit_assigned_context global_unit_assigned_context.units[1] -> unit unit = derived_unit derived_unit derived_unit.elements[2] -> derived_unit_element derived_unit_element.unit -> named_unit
DATE	calendar_date	41		
DATE_AND_TIME	1: (calendar_date) 2: (date_and_time)	41 41		
DESIGN_REQUEST	versioned_action_request	41		{versioned_action_request <- action_directive.requests[i] action_directive (action_directive.name = 'request for change') (action_directive.name = 'request for clarification')}
description	versioned_action_ request.description	41		
design_request to date_and_time (as request_date)	PATH		6	versioned_action_request 1: (date_assigned_item = versioned_action_request date_assigned_item <- casting_date_assignment.items[i] casting_date_assignment <= date_assignment {date_assignment date_assignment.role -> date_role date_role.name = 'request date'} date_assignment.assigned_date -> date => calendar_date) 2: (date_and_time_assigned_item = versioned_action_request

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
				date_and_time_assigned_item <- casting_date_and_time_assignment.items[i] casting_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'request date'} date_and_time_assignment.assigned_date_and_time -> date_and_time)
design_request to item_definition (as referenced_item)	PATH see note 3			 versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => casting_action_request_assignment casting_action_request_assignment.items[i] -> action_request_assigned_item (action_request_assigned_item = product_definition product_definition) (action_request_assigned_item = product_definition product_definition => casting_part) (action_request_assigned_item = product_definition product_definition => casting_mold_design_element) (action_request_assigned_item = product_definition product_definition => die_mold) (action_request_assigned_item = product_definition product_definition => investment_mold) (action_request_assigned_item = product_definition product_definition => investment_mold_pattern) (action_request_assigned_item = product_definition product_definition => investment_mold_sprue) (action_request_assigned_item = product_definition product_definition =>

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
				sand_mold) (action_request_assigned_item = property_definition property_definition) (action_request_assigned_item = shape_aspect shape_aspect)
design_request to person_and_organization (as requested_by)	PATH		7	versioned_action_request person_and_organization_assigned_item = versioned_action_request person_and_organization_assigned_item <- casting_person_and_organization_assignment.items[i] casting_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'requestor'} person_and_organization_assignment.assigned_person_- and_organization -> person_and_organization
DESIGN_UPDATE	directed_action	41		{directed_action <= executed_action <= action action.name = 'design update'}
description	action.description	41		directed_action <= executed_action <= action action.description
design_update to date_and_time (as action_date)	PATH		8	directed_action 1: (date_assigned_item = directed_action date_assigned_item <- casting_date_assignment.items[i] casting_date_assignment <= date_assignment {date_assignment date_assignment.role -> date_role date_role.name = 'update date'} date_assignment.assigned_date -> date => calendar_date) 2: (date_and_time_assigned_item = directed_action

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
				date_and_time_assigned_item <- casting_date_and_time_assignment.items[i] casting_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'update date'} date_and_time_assignment.assigned_date_and_time -> date_and_time)
design_update to design_request (as referenced)	PATH			directed_action directed_action.directive -> action_directive action_directive.requests[i] -> versioned_action_request
design_update to item_definition (as updated_item)	PATH see note 3		 9	 directed_action <= executed_action <= action <- action_assignment.assigned_action action_assignment => casting_action_assignment casting_action_assignment.items[i] -> action_assigned_item (action_assigned_item = product_definition product_definition) (action_assigned_item = product_definition product_definition => casting_part) (action_assigned_item = product_definition product_definition => casting_mold_design_element) (action_assigned_item = product_definition product_definition => die_mold) (action_assigned_item = product_definition product_definition =>

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
				investment_mold) (action_assigned_item = product_definition product_definition => investment_mold_pattern) (action_assigned_item = product_definition product_definition => investment_mold_sprue) (action_assigned_item = product_definition product_definition => sand_mold) (action_assigned_item = characterized_object characterized_object) (action_assigned_item = property_definition property_definition) (action_assigned_item = shape_aspect shape_aspect)
design_update to person_and_organization (as action_taker)	PATH		10	directed_action person_and_organization_assigned_item = directed_action person_and_organization_assigned_item <- casting_person_and_organization_assignment.items[i] casting_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'updater'} person_and_organization_assignment.assigned_person_- and_organization -> person_and_organization
DOCUMENT	document	41		
name	document.name	41		
GRAPHICS	document	41		{document document.kind -> document_type document_type.product_data_type = 'graphics'}

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
name	document.name	41		
ITEM	product	41		
description	product.description	41		
identification	product.id	41		
name	product.name	41		
ITEM_VERSION	product_definition_formation	41		

purchase_order_numbers	versioned_action_request.id	41		`product_definition_formation`
				`action_request_assigned_item =`
				`product_definition_formation`
				`action_request_assigned_item <-`
				`casting_action_request_assignment.items[i]`
				`casting_action_request_assignment <=`
				`action_request_assignment`
				`action_request_assignment.assigned_action_request ->`
				`versioned_action_request`
				`versioned_action_request.id`

item_version to item
(as associated_item)
	PATH			`product_definition_formation`
				`product_definition_formation.of_product ->`
				`product`

ORGANIZATION	organization	41		
PERSON	person	41		
PERSON_AND_ ORGANIZATION	person_and_organization	41		

PICTURE	document	41		`{document`
				`document.kind ->`
				`document_type`
				`document_type.product_data_type = 'picture'}`

title	document.id	41		

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
PROCESS_PLAN_VERSION	process_plan_version	223		process_plan_version <= action
process_plan_version to process_plan (as associated_plan)	IDENTICAL MAPPING			
REQUEST_FOR_CHANGE	versioned_action_request	41		{versioned_action_request <- action_directive.requests[i] action_directive action_directive.name = 'request for change'}
request_for_change to item_definition (as suggested_change)	PATH see note 2			 versioned_action_request <- action_request_solution.request action_request_solution action_request_solution.method -> action_method <- action.chosen_method {action action.name = 'suggest change'} action <- action_assignment.assigned_action action_assignment => casting_action_assignment casting_action_assignment.items[i] -> action_assigned_item (action_assigned_item = product_definition product_definition) (action_assigned_item = product_definition product_definition => casting_part) (action_assigned_item = product_definition product_definition => casting_mold_design_element) (action_assigned_item = product_definition product_definition => die_mold) (action_assigned_item = product_definition product_definition => investment_mold)

Table 5 – Continued

Application element	AIM element	Source	Rules	Reference path
				(action_assigned_item = product_definition product_definition => investment_mold_pattern) (action_assigned_item = product_definition product_definition => investment_mold_sprue) (action_assigned_item = product_definition product_definition => sand_mold)
REQUEST_FOR_ CLARIFICATION	versioned_action_request	41		{versioned_action_request <- action_directive.requests[i] action_directive action_directive.name = 'request for clarification'}
SPECIFICATION	document	41		{document document.kind -> document_type document_type.product_data_type = 'specification'}
specification_identification	document.id	41		
specification to organization (as issuing_organization)	PATH		16	document organization_assigned_item = document organization_assigned_item <- casting_organization_assignment.items[i] casting_organization_assignment <= organization_assignment {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'issuing organization'} organization_assignment.assigned_organization -> organization
VERSION	(product_definition_formation) (process_plan_version)	41 223		 process_plan_version <= action
description	(product_definition_ formation.description)	41		

Table 5 – Concluded

Application element	AIM element	Source	Rules	Reference path
	(product_definition_formation.description)	41		
identification	(product_definition_formation.id) NO MAPPING	41		

Table 6 – Mapping table for product_specification UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_ITEM	(casting_mold_design_element)(die_mold)(investment_mold_pattern)(investment_mold_sprue)(sand_mold)	223223223223223		(casting_mold_design_element <= product_definition {product_definition (product_definition.description = ‘pattern and rigging assembly’) (product_definition.description = ‘die sub-assembly’) (product_definition.description = ‘investment pattern’)}) (die_mold_definition <= product_definition) (investment_mold_pattern <= product_definition) (investment_mold_sprue <= product_definition) (sand_mold <= product_definition)
assembly_item to assembly_item_placement (as elements)	PATH			(casting_mold_design_element <=) (die_mold <=) (investment_mold_pattern <=) (investment_mold_sprue <=) (sand_mold <=) product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => mapped_item

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_ITEM_ PLACEMENT	mapped_item	43		
assembly_item_placement to axis_placement (as location)	PATH			mapped_item [mapped_item.mapping_target ->] [mapped_item.mapping_source -> representation_map representation_map.mapping_origin ->] representation_item => geometric_representation_item => placement
assembly_item_placement to item_definition (as placed_item)	PATH			mapped_item mapped_item.source -> representation_map representation_map.mapped_representation -> {representation => shape_representation} representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition (characterized_definition = characterized_object characterized_object) (characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition (product_definition) (product_definition => casting_part) (product_definition => casting_mold_design_element) (product_definition => die_mold) (product_definition => investment_mold) (product_definition =>

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				investment_mold_pattern) (product_definition => investment_mold_sprue) (product_definition => sand_mold))
CASTING_PART	casting_part	223	2	casting_part <= product_definition {product_definition [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'casting'] [product_definition.frame_of_reference -> {product_definition_context product_definition_context.life_cycle_stage = 'design'} product_definition_context <= application_context_element [application_context_element.name = 'foundry part'] [application_context_element.frame_of_reference -> application_context application_context.application = 'casting']]}
COMPOSITION_REQUIREMENT	material_property	45		{material_property <= property_definition property_definition.description = 'composition requirement'}
alloy_grade	descriptive_representation_item.description	45		material_property <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] ->

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				{representation_item representation_item.name = 'alloy grade'} representation_item => descriptive_representation_item descriptive_representation_item.description
composition_requirement to substance (as made_of)	PATH			material_property <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition product_definition {product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'material'}
CONSUMABLE_ITEM	(product_definition) (characterized_object)	41 41		
CORE	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = 'core'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'die casting') (application_context_element.name = 'investment casting') (application_context_element.name = 'sand casting')]] [product_definition <- product_definition_relationship.related_product_-definition {product_definition_relationship => product_definition_usage =>

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				assembly_component_usage} product_definition_relationship product_definition_relationship.relating_product_- definition -> (product_definition => die_mold) ({product_definition product_definition.description = 'investment pattern'} product_definition => casting_mold_design_element) (product_definition => sand_mold)]}
type_of_core	product_category.name	41		casting_mold_design_element <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name
core to core_master (as construction)	PATH			casting_mold_design_element <= product_definition <- product_definition_relationship.relating_product_- definition {[product_definition_relationship => product_definition_usage] [product_definition_relationship product_definition_relationship.name = 'definition usage']} product_definition_relationship product_definition_relationship.related_product_- definition -> product_definition {product_definition product_definition.description = 'core master'}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
CORE_MASTER	product_definition	41		{[product_definition [product_definition.description = ‘core master’] [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = ‘die casting’) (application_context_element.name = ‘investment casting’) (application_context_element.name = ‘sand casting’)]] [product_definition <- product_definition_relationship.relating_product_- definition {[product_definition_relationship => product_definition_usage] [product_definition_relationship product_definition_relationship.name = ‘definition usage’]} product_definition_relationship product_definition_relationship.related_product_- definition -> {product_definition product_definition.description = ‘core’} product_definition => casting_mold_design_element]}
CUSTOMER_PART	casting_part	223	2	casting_part <= product_definition {product_definition [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = ‘casting’] [product_definition.frame_of_reference -> {product_definition_context product_definition_context.life_cycle_stage = ‘design’}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition_context <= application_context_element [application_context_element.name = 'customer part'] [application_context_element.frame_of_reference -> application_context application_context.application = 'casting']]}
DIE_COMPONENT	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = 'die component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'die casting']] [product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.relating_product_- definition -> {product_definition product_definition.description = 'die sub-assembly'} product_definition => casting_mold_design_element]}
type_of_component	product_category.name	41		casting_mold_design_element <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
DIE_MOLD_ASSEMBLY	die_mold	223		die_mold <= product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'die casting'}
die_mold_assembly to shape_ representation (as cavity_shape)	PATH			die_mold <= product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition {property_definition => product_definition_shape} property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> {representation representation.name = 'die cavity'} representation => shape_representation => (advanced_brep_shape_representation) (faceted_brep_shape_representation) (elementary_brep_shape_representation) (geometrically_bounded_2d_shape_representation)
DIE_SUB_ASSEMBLY	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = 'die sub-assembly'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				application_context_element.name = 'die casting']] [product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.relating_product_- definition -> (product_definition => die_mold) ({product_definition product_definition.description = 'die sub-assembly'} product_definition => casting_mold_design_element)]}
type_of_sub_assembly	product_category.name	41		casting_mold_definition_element <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name
FINISHED_CASTING	casting_part	223	2	casting_part <= product_definition {product_definition [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'casting'] [product_definition.frame_of_reference -> {product_definition_context product_definition_context.life_cycle_stage = 'design'}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition_context <= application_context_element [application_context_element.name = ‘finished’] [application_context_element.frame_of_reference -> application_context application_context.application = ‘casting’]]}
FLASK	casting_mold_design_element	223		casting_mold_design_element <= product_definition {product_definition [product_definition.description = ‘flask’] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = ‘sand casting’]}
type_of_flask	product_category.name	41		casting_mold_design_element <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name
flask to measure_with_unit (as cope_height)	PATH			casting_mold_design_element <= product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation} representation

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation.items[i] -> {representation_item representation_item.name = ‘cope height’} representation_item => measure_representation_item <= {measure_with_unit => length_measure_with_unit} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
flask to measure_with_unit (as drag_height)	PATH			casting_mold_design_element <= product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation} representation representation.items[i] -> {representation_item representation_item.name = ‘drag height’} representation_item => measure_representation_item <= {measure_with_unit => length_measure_with_unit} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
flask to measure_with_unit (as flask_length)	PATH			casting_mold_design_element <= product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation} representation representation.items[i] -> {representation_item representation_item.name = 'flask height'} representation_item => measure_representation_item <= {measure_with_unit => length_measure_with_unit} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
flask to measure_with_unit (as flask_width)	PATH			casting_mold_design_element <= product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation} representation

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation.items[i] -> {representation_item representation_item.name = ‘flask width’} representation_item => measure_representation_item <= {measure_with_unit => length_measure_with_unit} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
GEOMETRY_REQUIREMENT	property_definition	41		{property_definition [property_definition.description = ‘shape requirement’] [property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition]}
geometry_requirement to shape_representation (as shape)	PATH			product_definition_shape <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => shape_representation => (advanced_brep_shape_representation) (faceted_brep_shape_representation) (elementary_brep_shape_representation) (geometrically_bounded_2d_shape_representation)
HEAT_TREAT_REQUIREMENT	property_definition	41		{property_definition property_definition.description = ‘heat treat process requirement’}
heat_treat_requirement to data_curve (as temperature_versus_time)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = ‘heat treat temperature profile’} representation =>

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				data_curve_representation
IN_MOLD_RIGGING_COMPONENT	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = 'in-mold rigging component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'die casting') (application_context_element.name = 'investment casting') (application_context_element.name = 'sand casting')]] [product_definition <- product_definition_relationship.related_product_definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.relating_product_definition -> product_definition => (die_mold) (investment_mold_sprue) (sand_mold)]}
type_of_rigging	product_category.name	41		casting_mold_design_element <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
INSERT	casting_mold_design_element	223		casting_mold_design_element <= {[product_definition [product_definition.description = 'insert'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'die casting') (application_context_element.name = 'investment casting') (application_context_element.name = 'sand casting')]] [product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.relating_product_- definition -> (product_definition => die_mold) ({product_definition product_definition.description = 'investment pattern'} product_definition => casting_mold_design_element) (product_definition => sand_mold)]}
INSPECTION_OR_TEST_ REQUIREMENT	property_inspection_or_test_ requirement	223		property_inspection_or_test_requirement <= property_definition
sampling_run_size	measure_with_unit.value_ component	41		property_inspection_or_test_requirement <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] ->

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				{representation_item representation_item.name = 'sample run size'} representation_item => measure_representation_item <= measure_with_unit measure_with_unit.value_component {measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
test_name	property_definition.name	45		property_inspection_or_test_requirement <= property_definition property_definition.name
inspection_or_test_requirement to item_requirement (as inspection_against)	PATH			property_inspection_or_test_requirement <= property_definition <- property_definition_relationship.related_property_- definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'inspection or test basis'} property_definition_relationship.relating_property_- definition -> (property_definition) (property_definition => property_inspection_or_test_requirement) (property_definition => reporting_requirement) (property_definition => material_property) (property_definition {property_definition property_definition.description = 'tolerance requirement'} property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				(shape_definition = shape_aspect shape_aspect <- dimensional_size.applies_to dimensional_size) (shape_definition = shape_aspect shape_aspect <- geometric_tolerance.toleranced_shape_aspect geometric_tolerance) (shape_definition = shape_aspect_relationship shape_aspect_relationship => dimensional_location))
inspection_or_test_requirement to measure_with_unit (as frequency)	PATH			property_inspection_or_test_requirement <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item (representation_item.name = 'inspection frequency') (representation_item.name = 'test frequency')} representation_item => measure_representation_item <= {measure_with_unit => ratio_measure_with_unit} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
INVESTMENT_MOLD	investment_mold	223		investment_mold <= product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'investment casting'}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
investment_mold_type	product_category.name	41		investment_mold <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name
investment_mold to shape_representation (as cavity_shape)	PATH			investment_mold <= product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition {property_definition => product_definition_shape} property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> {representation representation.name = 'investment mold cavity'} representation => shape_representation => (advanced_brep_shape_representation) (faceted_brep_shape_representation) (elementary_brep_shape_representation) (geometrically_bounded_2d_shape_representation)
INVESTMENT_PATTERN_ ASSEMBLY	investment_mold_pattern	223		investment_mold_pattern <= product_definition {[product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'investment casting']

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				[product_definition <- product_definition_relationship.related_product_- definition product_definition_relationship {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship.relating_product_- definition -> product_definition => investment_mold]}
INVESTMENT_PATTERN	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = 'investment pattern'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'investment casting']] [product_definition <- product_definition_relationship.related_product_- definition product_definition_relationship {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship.relating_product_- definition -> product_definition => investment_mold_pattern]}
investment_pattern to investment_pattern_die (as construction)	PATH			casting_mold_design_element <= product_definition <- product_definition_relationship.relating_product_- definition product_definition_relationship {[product_definition_relationship => product_definition_usage] [product_definition_relationship

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.name = ‘construction’]} product_definition_relationship.related_product_-definition -> product_definition => die_mold}
INVESTMENT_PATTERN_DIE_COMPONENT	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [(product_definition.description = ‘die component’) (product_definition.description = ‘die sub-assembly’)] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = ‘investment casting’]] [product_definition <- product_definition_relationship.related_product_-definition product_definition_relationship {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship.relating_product_-definition -> product_definition => die_mold]}
INVESTMENT_PATTERN_DIE	die_mold	223		die_mold <= product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = ‘investment casting’}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
INVESTMENT_SPRUE_ASSEMBLY	investment_mold_sprue_definition	223		investment_mold_sprue <= product_definition {[product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'investment casting'] [product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship.relating_product_definition -> product_definition => investment_mold_pattern
INVESTMENT_SPRUE_COMPONENT	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = 'sprue component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'investment casting']] [product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship.relating_product_definition -> product_definition => investment_mold_sprue]}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
ITEM_DEFINITION	(characterized_object)	41		
	(product_definition)	41		
	(casting_part)	223		
	(casting_mold_design_element)	223		
	(die_mold)	223		
	(investment_mold)	223		
	(investment_mold_pattern)	223		
	(investment_mold_sprue)	223		
	(sand_mold)	223	2	(casting_part <=) (casting_mold_design_element <=) (die_mold <=) (investment_mold <=) (investment_mold_pattern <=) (investment_mold_sprue <=) (sand_mold <=) product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.frame_of_reference -> application_context application_context.application = 'casting'}
description	(characterized_object.description)(product_definition.description)	4141		(casting_part <=) (casting_mold_design_element <=) (die_mold <=) (investment_mold <=) (investment_mold_pattern <=) (investment_mold_sprue <=) (sand_mold <=)

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition product_definition.description
identification	(NO MAPPING)(product_definition.id)	41		(casting_part <=) (casting_mold_design_element <=) (die_mold <=) (investment_mold <=) (investment_mold_pattern <=) (investment_mold_sprue <=) (sand_mold <=) product_definition product_definition.id
name	(characterized_object.name)(descriptive_representation_item.description)	4145		(casting_part <= product_definition) (casting_mold_design_element <= product_definition) (die_mold <= product_definition) (investment_mold <= product_definition) (investment_mold_pattern <= product_definition) (investment_mold_sprue <= product_definition) (sand_mold <= product_definition) (product_definition) characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'item name'}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation_item => descriptive_representation_item descriptive_representation_item.description
item_definition to item_requirement (as requirements)	PATH			((casting_part <= product_definition) (casting_mold_design_element <= product_definition) (die_mold <= product_definition) (investment_mold <= product_definition) (investment_mold_pattern <= product_definition) (investment_mold_sprue <= product_definition) (sand_mold <= product_definition) (product_definition) characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition) (characterized_object characterized_definition = characterized_object) characterized_definition <- property_definition.definition (property_definition) (property_definition => property_inspection_or_testing_requirement) (property_definition => reporting_requirement) (property_definition => product_definition_shape shape_definition = product_definition_shape shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition)

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				({property_definition property_definition.description = 'tolerance requirement'} property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- dimensional_size.applies_to dimensional_size) ({property_definition property_definition.description = 'tolerance requirement'} property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- geometric_tolerance.toleranced_shape_aspect geometric_tolerance) ({property_definition property_definition.description = 'tolerance requirement'} property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- [shape_aspect_relationship.relating_shape_aspect] [shape_aspect_relationship.related_shape_aspect] shape_aspect_relationship => dimensional_location)
item_definition to item_version (as associated_item_version)	PATH			(casting_part <= product_definition) (casting_mold_design_element <= product_definition) (die_mold <= product_definition) (investment_mold <=

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition) (investment_mold_pattern <= product_definition) (investment_mold_sprue <= product_definition) (sand_mold <= product_definition) (product_definition) product_definition.formation -> product_definition_formation
item_definition to person_and_organization (as designer)	PATH			(characterized_object person_and_organization_assigned_item = characterized_object) (casting_part <= product_definition person_and_organization_assigned_item = product_definition) (casting_mold_design_element <= product_definition person_and_organization_assigned_item = product_definition) (die_mold <= product_definition person_and_organization_assigned_item = product_definition) (investment_mold <= product_definition person_and_organization_assigned_item = product_definition) (investment_mold_pattern <= product_definition person_and_organization_assigned_item = product_definition) (investment_mold_sprue <= product_definition person_and_organization_assigned_item = product_definition) (sand_mold <=

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition person_and_organization_assigned_item = product_definition) (product_definition person_and_organization_assigned_item = product_definition) person_and_organization_assigned_item <- casting_person_and_organization_assignment.items[i] casting_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'designer'} person_and_organization_assignment.assigned_person_- and_organization -> person_and_organization
ITEM_REQUIREMENT	(property_definition)	41		
	(property_inspection_or_test_requirement)	223		
	(reporting_requirement)	223		
	(material_property)	45		
	(geometric_tolerance)	47		
	(dimensional_size)	47		
	(dimensional_location)	47	11	(property_inspection_or_test_requirement <=) (reporting_requirement <=) property_definition
description	descriptive_representation_item.description	45		(property_inspection_or_test_requirement <= property_definition <-) (reporting_requirement <= property_definition <-) (material_property <= property_definition <-) ((geometric_tolerance

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				geometric_tolerance.toleranced_shape_aspect -> shape_aspect shape_definition = shape_aspect) (dimensional_size dimensional_size.applies_to -> shape_aspect shape_definition = shape_aspect) (dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship) shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <-) (property_definition <-) property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
item_requirement to picture (as illustration)	PATH			(property_definition document_assigned_item = property_definition) (property_inspection_or_test_requirement <= property_definition document_assigned_item = property_definition) (reporting_requirement <= property_definition document_assigned_item = property_definition) (material_property <= property_definition document_assigned_item = property_definition) (geometric_tolerance document_assigned_item = geometric_tolerance) (dimensional_size

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				document_assigned_item = dimensional_size) (dimensional_location document_assigned_item = dimensional_location) document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type.product_data_type = 'picture'}
item_requirement to shape_aspect (as applies_to)	PATH			((property_inspection_or_test_requirement <= property_definition) (reporting_requirement <= property_definition) (material_property <= property_definition) (property_definition) property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect) (geometric_tolerance geometric_tolerance.toleranced_shape_aspect ->) (dimensional_size dimensional_size.applies_to ->) (dimensional_location <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect ->] [shape_aspect_relationship.related_shape_aspect ->]) shape_aspect
item_requirement to specification_reference (as referenced_from)	PATH		12	(property_definition document_assigned_item = property_definition) (property_inspection_or_test_requirement <= property_definition document_assigned_item = property_definition) (reporting_requirement <= (property_definition

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				document_assigned_item = property_definition) (material_property <= property_definition document_assigned_item = property_definition) (geometric_tolerance document_assigned_item = geometric_tolerance) (dimensional_size document_assigned_item = dimensional_size) (dimensional_location document_assigned_item = dimensional_location) document_assigned_item <- casting_document_reference.items[i] casting_document_reference {casting_document_reference <= document_reference document_reference.assigned_document -> document document.kind -> document_type document_type.product_data_type = ‘specification’}
MASTER	product_definition	41		{[product_definition [(product_definition.description = ‘core master’) (product_definition.description = ‘pattern master’)] [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = ‘die casting’) (application_context_element.name = ‘investment casting’) (application_context_element.name = ‘sand casting’)]] [product_definition <- product_definition_relationship.related_product_- definition {[product_definition_relationship => product_definition_usage] [product_definition_relationship

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.name = ‘definition usage’]} product_definition_relationship product_definition_relationship.relating_product_-definition -> {product_definition (product_definition.description = ‘core’) (product_definition.description = ‘investment pattern’) (product_definition.description = ‘pattern’)} product_definition => casting_mold_design_element]}
storage_location	descriptive_representation_item.description	45		product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = ‘storage location’} representation_item => descriptive_representation_item descriptive_representation_item.description
PATTERN_AND_RIGGING	casting_mold_design_element	223		casting_mold_design_element <= product_definition {product_definition [product_definition.description = ‘pattern and rigging assembly’] [product_definition.frame_of_reference -> product_definition_context <= application_context_element

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				application_context_element.name = 'sand casting']}
PATTERN	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = 'pattern'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'sand casting']] [product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.relating_product_- definition -> {product_definition product_definition.description = 'pattern and rigging assembly'} product_definition => casting_mold_design_element]}
pattern to pattern_master (as construction)	PATH			casting_mold_design_element <= product_definition <- product_definition_relationship.relating_product_- definition {[product_definition_relationship => product_definition_usage] [product_definition_relationship product_definition_relationship.name = 'definition usage']} product_definition_relationship product_definition_relationship.related_product_- definition -> product_definition {product_definition product_definition.description = 'pattern master'}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
PATTERN_MASTER	product_definition	41		{[product_definition [product_definition.description = ‘pattern master’] [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = ‘die casting’) (application_context_element.name = ‘investment casting’) (application_context_element.name = ‘sand casting’)]] [product_definition <- product_definition_relationship.related_product_- definition {[product_definition_relationship => product_definition_usage] [product_definition_relationship product_definition_relationship.name = ‘definition usage’]} product_definition_relationship product_definition_relationship.relating_product_- definition -> {product_definition (product_definition.description = ‘pattern’) (product_definition.description = ‘investment pattern’)} product_definition => casting_mold_design_element]}
PATTERN_PLATE	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = ‘pattern plate’] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = ‘sand casting’]] [product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.relating_product_- definition -> {product_definition product_definition.description = 'pattern and rigging assembly'} product_definition => casting_mold_design_element]}
type_of_plate	product_category.name	41		casting_mold_design_element <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name
PATTERN_RIGGING_ COMPONENT	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = 'rigging component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'sand casting']] [product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.relating_product_- definition -> {product_definition product_definition.description = 'pattern and rigging assembly'} product_definition => casting_mold_design_element]}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
type_of_rigging	product_category.name	41		casting_mold_design_element <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name
PROCESS_REQUIREMENT	property_definition	41		{property_definition (property_definition.description = 'process requirement') (property_definition.description = 'heat treat process requirement')}
process_name	property_definition.name	41		
PROPERTY_REQUIREMENT	property_definition	41		{property_definition property_definition.description = 'property requirement'}
property_requirement to property_relationship (as definition)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => data_curve_representation
property_requirement to property_value (as definition)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => measure_representation_item

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
RAW_CASTING	casting_part_definition	223	2	casting_part <= product_definition {product_definition [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'casting'] [product_definition.frame_of_reference -> {product_definition_context product_definition_context.life_cycle_stage = 'design'} product_definition_context <= application_context_element [application_context_element.name = 'raw'] [application_context_element.frame_of_reference -> application_context application_context.application = 'casting']]}
REPORTING_ REQUIREMENT	reporting_requirement	223		reporting_requirement <= property_definition {property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition product_definition => casting_part}
data_to_be_reported	descriptive_representation_ item.description	45		reporting_requirement <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] ->

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation_item => descriptive_representation_item descriptive_representation_item.description
reporting_requirement to item_requirement (as based_on)	PATH			reporting_requirement <= property_definition <- property_definition_relationship.related_property_-definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'reporting basis'} property_definition_relationship.relating_property_-definition -> (property_definition) (property_definition => reporting_requirement) (property_definition => material_property) ({property_definition property_definition.description = 'tolerance requirement'} property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition (shape_definition = shape_aspect shape_aspect <- geometric_tolerance.toleranced_shape_aspect geometric_tolerance) (shape_definition = shape_aspect shape_aspect <- dimensional_size.applies_to dimensional_size) (shape_definition = shape_aspect_relationship shape_aspect_relationship => dimensional_location))

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
REQUIRED_MACHINING_ ALLOWANCE	property_definition	41		{property_definition property_definition.description = ‘required machining allowance’}
required_machining_allowance to measure_range_with_unit (as value)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier (type_qualifier.name = ‘nominal amount’) (type_qualifier.name = ‘lower limit’) (type_qualifier.name = ‘upper limit’)} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
SAND_MOLD_ASSEMBLY	sand_mold	223		sand_mold <= product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = ‘sand casting’}
SAND_MOLD_COMPONENT	casting_mold_design_element	223		casting_mold_design_element <= product_definition {[product_definition [product_definition.description = ‘sand mold component’] [product_definition.frame_of_reference -> product_definition_context <= application_context_element

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				application_context_element.name = ‘sand casting’]] [product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship => product_definition_usage => assembly_component_usage} {product_definition_relationship (product_definition_relationship.name = ‘cope’) (product_definition_relationship.name = ‘drag’)} product_definition_relationship product_definition_relationship.relating_product_- definition -> product_definition => sand_mold]}
sand_mold_component to flask (as construction_flask)	PATH			casting_mold_design_element <= product_definition <- product_definition_relationship.relating_product_- definition product_definition_relationship {[product_definition_relationship => product_definition_usage] [product_definition_relationship product_definition_relationship.name = ‘construction’]} product_definition_relationship.related_product_- definition -> {product_definition product_definition.description = ‘flask’} product_definition => casting_mold_design_element
sand_mold_component to pattern_and_rigging (as construction_pattern_and_- rigging)	PATH			casting_mold_design_element <= product_definition <- product_definition_relationship.relating_product_- definition product_definition_relationship {[product_definition_relationship => product_definition_usage] [product_definition_relationship

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.name = ‘construction’]} product_definition_relationship.related_product_- definition -> {product_definition product_definition.description = ‘pattern and rigging assembly’ product_definition => casting_mold_design_element
SAND_MOLD	sand_mold	223		sand_mold <= product_definition {[product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = ‘sand casting’] [product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.relating_product_- definition -> product_definition => sand_mold]}
sand_mold to shape_representation (as cavity_shape)	PATH			sand_mold <= product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition {property_definition => product_definition_shape} property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation}

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> {representation representation.name = ‘sand mold cavity’} representation => shape_representation => (advanced_brep_shape_representation) (faceted_brep_shape_representation) (elementary_brep_shape_representation) (geometrically_bounded_2d_shape_representation)
SECONDARY_TOOLING	product_definition	41		{product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = ‘die casting’) (application_context_element.name = ‘investment casting’) (application_context_element.name = ‘sand casting’)] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= (product_category) (product_category <- (product_category_relationship.sub_category) (product_category_relationship.category) product_category_relationship {product_category_relationship } (product_category_relationship.sub_category ->) (product_category_relationship.category ->) product_category) product_category.name = ‘tooling’]}
type_of_tooling	product_category.name	41		product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <=

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_category product_category.name
SIMPLE_ITEM	(casting_part)	223		
	(casting_mold_design_element)	223		
	(investment_mold)	223		
	(product_definition)	41		(casting_part <= product_definition) (casting_mold_design_element <= product_definition {product_definition (product_definition.description = 'core') (product_definition.description = 'die component') (product_definition.description = 'die sub-assembly') (product_definition.description = 'flask') (product_definition.description = 'in-mold rigging component') (product_definition.description = 'insert') (product_definition.description = 'pattern') (product_definition.description = 'pattern plate') (product_definition.description = 'rigging component') (product_definition.description = 'sand mold component') (product_definition.description = 'sprue component')}) (investment_mold <= product_definition) ({product_definition (product_definition.description = 'core master') (product_definition.description = 'pattern master') (product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= (product_category)

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				(product_category <- (product_category_relationship.sub_category) (product_category_relationship.category) product_category_relationship {product_category_relationship } (product_category_relationship.sub_category ->) (product_category_relationship.category ->) product_category) product_category.name = ‘tooling’)})
SPECIFICATION_ REFERENCE	casting_document_reference	223		casting_document_reference <= document_reference {document_reference document_reference.assigned_document -> document document.kind -> document_type document_type.product_data_type = ‘specification’}
designation	document_usage_ constraint.subject_element	41		casting_document_reference <= document_reference document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint document_usage_constraint.subject_element
exceptions	referenced_property_exception	223		casting_document_reference casting_document_reference.items[i] -> document_assigned_item document_assigned_item = property_definition property_definition <- property_definition_relationship.relating_property_- definition property_definition_relationship property_definition_relationship.related_property_- definition -> {property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] ->

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation_item => descriptive_representation_item} property_definition => referenced_property_exception
specification_reference to specification (as referenced_specification)	PATH			casting_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type.product_data_type = 'specification'}
SURFACE_ROUGHNESS_REQUIREMENT	property_definition	41		{property_definition property_definition.description = 'surface roughness requirement'}
surface_roughness_requirement to curve_on_surface (as measuring_direction)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => geometric_representation_item => curve => surface_curve
surface_roughness_requirement to measure_range_with_unit (as amount)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier (type_qualifier.name = 'nominal amount') (type_qualifier.name = 'lower limit')

Table 6 – Continued

Application element	AIM element	Source	Rules	Reference path
				(type_qualifier.name = 'upper limit')} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
UPDATE_RELATIONSHIP	(product_definition_relationship)(change_from_assignment)(change_to_assignment)	41223223		({product_definition_relationship product_definition_relationship.name = 'definition update'}) (change_from_assignment <= action_assignment) (change_to_assignment <= action_assignment)
update_relationship to item_definition (as original)	PATH			(product_definition_relationship product_definition_relationship.relating_product_-definition -> (product_definition) (product_definition => casting_part) (product_definition => casting_mold_design_element) (product_definition => die_mold) (product_definition => investment_mold) (product_definition => investment_mold_pattern) (product_definition => investment_mold_sprue) (product_definition => sand_mold)) (change_from_assignment change_from_assignment.items[i] -> change_item change_item = characterized_object)
update_relationship to item_definition (as updated)	PATH			(product_definition_relationship product_definition_relationship.related_product_-definition -> (product_definition) (product_definition => casting_part) (product_definition => casting_mold_design_element)

Table 6 – Concluded

Application element	AIM element	Source	Rules	Reference path
				(product_definition => die_mold) (product_definition => investment_mold) (product_definition => investment_mold_pattern) (product_definition => investment_mold_sprue) (product_definition => sand_mold)) (change_to_assignment change_to_assignment.items[i] -> change_item change_item = characterized_object)

Table 7 – Mapping table for property UoF

Application element	AIM element	Source	Rules	Reference path
PROPERTY_ RELATIONSHIP	data_curve_representation	223		`data_curve_representation <=` `representation`

x_property_name	representation_item.name	43		`data_curve_representation <=`
				`representation`
				`representation.items[1] ->`
				`representation_item`
				`representation_item.name`

y_property_name	representation_item.name	43		`data_curve_representation <=`
				`representation`
				`representation.items[2] ->`
				`representation_item`
				`representation_item.name`

property_relationship to data_curve (as relationship_curve)	IDENTICAL MAPPING			
PROPERTY_VALUE	measure_representation_item	45		

name	representation_item.name	43		`measure_representation_item <=`
				`representation_item`
				`representation_item.name`

property_value to measure_range_with_unit
(as value)
	PATH			`measure_representation_item <=`
				`measure_with_unit`
				`[measure_with_unit.value_component]`
				`[measure_with_unit.unit_component]`

Table 8 – Mapping table for quality_assurance UoF

Application element	AIM element	Source	Rules	Reference path
ARTIFACT	casting_part	223	2	casting_part <= product_definition {product_definition [product_definition.description = 'artifact'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'casting'] [product_definition.frame_of_reference -> {product_definition_context product_definition_context.life_cycle_stage = 'production'} product_definition_context <= application_context_element [(application_context_element.name = 'raw') (application_context_element.name = 'finished')] [application_context_element.frame_of_reference -> application_context application_context.application = 'casting']]}
part_number	product.id	41		casting_part <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.id
artifact to date_and_time (as production_date)	PATH			casting_part 1: (date_assigned_item = casting_part date_assigned_item <- casting_date_assignment.items[i] casting_date_assignment <= date_assignment {date_assignment

Table 8 – Continued

Application element	AIM element	Source	Rules	Reference path
				date_assignment.role -> date_role date_role.name = ‘production date’} date_assignment.assigned_date -> date => calendar_date) 2: (date_and_time_assigned_item = casting_part date_and_time_assigned_item <- casting_date_and_time_assignment.items[i] casting_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = ‘production date’} date_and_time_assignment.assigned_date_and_time -> date_and_time)
artifact to heat (as belongs_to_heat)	PATH			casting_part effectivity_assigned_item = casting_part effectivity_assigned_item <- casting_effectivity_assignment.items[i] casting_effectivity_assignment <= effectivity_assignment effectivity_assignment.assigned_effectivity -> effectivity => heat_effectivity
artifact to item_definition (as production_specification_used)	PATHsee note 2			casting_part <= product_definition <- product_definition_relationship.related_product_- definition {product_definition_relationship product_definition_relationship.name = ‘production specification’} product_definition_relationship product_definition_relationship.relating_product_- definition -> (product_definition) (product_definition => casting_part) (product_definition => casting_mold_design_element) (product_definition => die_mold)

Table 8 – Continued

Application element	AIM element	Source	Rules	Reference path
				(product_definition => investment_mold) (product_definition => investment_mold_pattern) (product_definition => investment_mold_sprue) (product_definition => sand_mold)
artifact to lot (as belongs_to_lot)	PATH			casting_part effectivity_assigned_item = casting_part effectivity_assigned_item <- casting_effectivity_assignment.items[i] casting_effectivity_assignment <= effectivity_assignment effectivity_assignment.assigned_effectivity -> effectivity => lot_effectivity
artifact to process_plan (as process_plan_used)	PATH			casting_part action_assigned_item = casting_part action_assigned_item <- casting_action_assignment.items[i] casting_action_assignment <= action_assignment action_assignment.assigned_action -> action => process_plan_version
FAILURE_REPORT	representation	43		{representation representation.name = 'failure report'}
description	descriptive_representation_item.-	45		representation representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
number_failed	measure_with_unit.value_component	41		representation representation.items[i] -> {representation_item representation_item.name = 'quantity failed'} representation_item => measure_representation_item <= measure_with_unit

Table 8 – Continued

Application element	AIM element	Source	Rules	Reference path
				measure_with_unit.value_component {measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
failure_report to artifact (as failed_artifacts)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition {product_definition product_definition.description = 'artifact'} product_definition => casting_part
failure_report to inspection_or_test_result (as what_failed)	PATH			representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation {representation representation.name = 'inspection or test result'}
HEAT	heat_effectivity	223		heat_effectivity <= effectivity
heat_number	effectivity.id	41		heat_effectivity <= effectivity effectivity.id
heat to date_and_time (as production_date)	PATH			heat_effectivity 1: (date_assigned_item = heat_effectivity date_assigned_item <- casting_date_assignment.items[i] casting_date_assignment <= date_assignment {date_assignment

Table 8 – Continued

Application element	AIM element	Source	Rules	Reference path
				date_assignment.role -> date_role date_role.name = 'production date'} date_assignment.assigned_date -> date => calendar_date) 2: (date_and_time_assigned_item = heat_effectivity date_and_time_assigned_item <- casting_date_and_time_assignment.items[i] casting_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'production date'} date_and_time_assignment.assigned_date_and_time -> date_and_time)
INSPECTION_OR_TEST_RESULT	representation	43		{representation representation.name = 'inspection or test result'}
description	descriptive_representation_item.-	45		representation representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
result_label	representation_item.name	43		representation representation.items[i] -> {representation_item => descriptive_representation_item} representation_item representation_item.name
inspection_or_test_result to item_requirement (as requirement)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> (property_definition) (property_definition => property_inspection_or_test_requirement)

Table 8 – Continued

Application element	AIM element	Source	Rules	Reference path
				(property_definition => reporting_requirement) (property_definition => material_property) ({property_definition property_definition.description = 'tolerance requirement'} property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition (shape_definition = shape_aspect shape_aspect <- geometric_tolerance.toleranced_shape_aspect geometric_tolerance) (shape_definition = shape_aspect shape_aspect <- dimensional_size.applies_to dimensional_size) (shape_definition = shape_aspect_relationship shape_aspect_relationship => dimensional_location))
inspection_or_test_result to measure_with_unit (as result)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'result'} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
inspection_or_test_result to sampled_set (as belongs_to)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition

Table 8 – Continued

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition {product_definition product_definition.description = 'sampled set'} product_definition => casting_part
inspection_or_test_result to shape_aspect (as applies_to)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect
inspection_or_test_result to statistical_measure (as result)	PATH			representation <- representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'result'} representation_relationship.rep_2 -> representation => statistical_measure
inspection_or_test_result to substance (as result)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition product_definition {product_definition product_definition.formation -> product_definition_formation

Table 8 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'material'}
LOT	lot_effectivity	41		
lot_number	effectivity.id	41		lot_effectivity <= effectivity effectivity.id
lot to date_and_time (as production_date)	PATH		13	lot_effectivity 1: (date_assigned_item = lot_effectivity date_assigned_item <- casting_date_assignment.items[i] casting_date_assignment <= date_assignment {date_assignment date_assignment.role -> date_role date_role.name = 'production date'} date_assignment.assigned_date -> date => calendar_date) 2: (date_and_time_assigned_item = lot_effectivity date_and_time_assigned_item <- casting_date_and_time_assignment.items[i] casting_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'production date'} date_and_time_assignment.assigned_date_and_time -> date_and_time)
SAMPLED_SET	casting_part	223	2	casting_part <= product_definition {product_definition [product_definition.description = 'sampled set'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product ->

Table 8 – Continued

Application element	AIM element	Source	Rules	Reference path
				product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = ‘casting’] [product_definition.frame_of_reference -> {product_definition_context product_definition_context.life_cycle_stage = ‘production’} product_definition_context <= application_context_element [(application_context_element.name = ‘raw’) (application_context_element.name = ‘finished’)] [application_context_element.frame_of_reference -> application_context application_context.application = ‘casting’]]}
size	product_definition.description	41		casting_part <= product_definition product_definition.description
sampled_set to artifact (as artifacts_inspected)	PATH			casting_part <= product_definition <- product_definition_relationship.related_product_- definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = ‘inspected artifact’} product_definition_relationship.relating_product_- definition -> {product_definition product_definition.description = ‘artifact’} product_definition => casting_part
sampled_set to heat (as sampled_from)	PATH			casting_part effectivity_assigned_item = casting_part effectivity_assigned_item <- casting_effectivity_assignment.items[i] casting_effectivity_assignment <= effectivity_assignment effectivity_assignment.assigned_effectivity ->

Table 8 – Concluded

Application element	AIM element	Source	Rules	Reference path
				effectivity => heat_effectivity
sampled_set to lot (as sampled_from)	PATH			casting_part effectivity_assigned_item = casting_part effectivity_assigned_item <- casting_effectivity_assignment.items[i] casting_effectivity_assignment <= effectivity_assignment effectivity_assignment.assigned_effectivity -> effectivity => lot_effectivity

Table 9 – Mapping table for quality_control UoF

Application element	AIM element	Source	Rules	Reference path
MACHINE_SETTING_RECORD	representation	43		{representation representation.name = 'machine setting record'}
machine_setting_record to measure_with_unit (as actual_setting)	PATH			representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
machine_setting_record to equipment_usage (as equipment_usage_context)	PATHsee note 1			representation <- resource_property_representation.representation resource_property_representation resource_property_representation.property -> resource_property resource_property.resource -> characterized_resource_definition characterized_resource_definition = action_resource action_resource {action_resource action_resource.kind -> action_resource_type action_resource_type.name = 'equipment'}
PROCESS_EXECUTION	[executed_action] [(product_property_process) (product_definition_process)]	41 49 49		
description	action.description	41		[executed_action <=] [(product_property_process <=) (product_definition_process <=)] action action.description

Table 9 – Continued

Application element	AIM element	Source	Rules	Reference path
process_execution to process (as deviated_process)	PATH			(product_definition_process <=) (product_property_process <=) action action.chosen_method -> (action_method) (action_method <- action_method_relationship.relating_action_method action_method_relationship {action_method_relationship } {action_method_relationship => (concurrent_action_method) (serial_action_method) (serial_action_method => sequential_method)} action_method_relationship.related_action_method -> action_method)
process_execution to process_ plan (as plan_used)	PATH		15	[executed_action <=] [(product_property_process <=) (product_definition_process <=)] action <- action_relationship.related_action {action_relationship action_relationship.name = 'deviated process'} action_relationship action_relationship.relating_action -> action => process_plan_version
process_execution to sampled_ set (as parts_measured)	PATH			(product_property_process <- process_property_association.process process_property_association process_property_association.property -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition) (product_definition_process <- process_product_association.process process_product_association process_product_association.defined_product ->) characterized_product_definition characterized_product_definition = product_definition

Table 9 – Continued

Application element	AIM element	Source	Rules	Reference path
				{product_definition product_definition.description = 'sampled set'} product_definition => casting_part
PROCESS_EXECUTION_ RECORD	(representation) (representation_relationship)	43 43		 ({representation (representation.name = 'machine setting record') (representation.name = 'substance usage record') (representation.name = 'substance composition element record')}) ({representation_relationship representation_relationship.name = 'process parameter record'})
process_execution_record to date_and_time (as recording_date)	PATH			(representation) (representation_relationship representation_relationship.rep_2 -> representation) 1: (date_assigned_item = representation date_assigned_item <- casting_date_assignment.items[i] casting_date_assignment <= date_assignment {date_assignment date_assignment.role -> date_role date_role.name = 'record date'} date_assignment.assigned_date -> date => calendar_date) 2: (date_and_time_assigned_item = representation date_and_time_assigned_item <- casting_date_and_time_assignment.items[i] casting_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'record date'} date_and_time_assignment.assigned_date_and_time -> date_and_time

Table 9 – Continued

Application element	AIM element	Source	Rules	Reference path
process_execution_record to process (as process_context)	PATH		14	(representation <-) (representation_relationship [representation_relationship.rep_1 ->] [representation_relationship.rep_2 ->] representation <-) action_property_representation.representation action_property_representation action_property_representation.property -> action_property action_property.definition -> characterized_action_definition characterized_action_definition = action_method action_method
PROCESS_PARAMETER_RECORD	representation_relationship	43		{representation_relationship representation_relationship.name = 'process parameter record'}
process_parameter_record to property_relationship (as actual_parameter)	PATH			representation_relationship representation_relationship.rep_2 -> representation => data_curve_representation
process_parameter_record to property_relationship (as planned_parameter)	PATH			representation_relationship representation_relationship.rep_1 -> representation => data_curve_representation
process_parameter_record to property_value (as actual_parameter)	PATH			representation_relationship representation_relationship.rep_2 -> representation representation.items[i] -> representation_item => measure_representation_item
process_parameter_record to property_value (as planned_parameter)	PATH			representation_relationship representation_relationship.rep_1 -> representation representation.items[i] -> representation_item => measure_representation_item
SUBSTANCE_COMPOSITION_ELEMENT_RECORD	representation	43		{[representation representation.name = 'substance composition element record'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition

Table 9 – Continued

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship => product_material_composition_relationship]}
actual_substance_composition_ amount	[measure_with_unit.value_ component]	41		
	[measure_with_unit.unit_ component]	41		representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
substance_ composition_element_record to substance_usage (as element)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition_relationship {product_definition_relationship => product_material_composition_relationship} product_definition_relationship product_definition_relationship.relating_product_- definition ->

Table 9 – Continued

Application element	AIM element	Source	Rules	Reference path
				product_definition <- product_definition_relationship.relating_product_- definition {product_definition_relationship => product_definition_usage => make_from_usage_option} product_definition_relationship product_definition_relationship.related_product_- definition
SUBSTANCE_USAGE_ RECORD	representation	43		{[representation representation.name = ‘substance usage record] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship => product_definition_usage => make_from_usage_option]}
actual_substance_usage_ amount	[measure_with_unit.value_ component] [measure_with_unit.unit_ component]	41 41		 representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]

Table 9 – Concluded

Application element	AIM element	Source	Rules	Reference path

Table 10 – Mapping table for simulation UoF

Application element	AIM element	Source	Rules	Reference path
BOUNDARY_CONDITION	representation	43		{representation representation.name = 'boundary condition'}
boundary_condition to property_relationship (as condition_value)	PATH			representation => data_curve_representation
boundary_condition to property_value (as condition_value)	PATH			representation representation.items[i] -> representation_item => measure_representation_item
boundary_condition to shape_aspect (as applies_to)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect
CONDITION_OR_ASSUMPTION	representation	43		{representation (representation.name = 'condition') (representation.name = 'assumption')}
description	descriptive_representation_item.-	45		representation representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
condition_or_assumption to document (as further_information)	PATH			representation document_assigned_item = representation document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document ->

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
				document
condition_or_assumption to property_relationship (as associated_properties)	PATH			representation => data_curve_representation
condition_or_assumption to property_value(as associated_-properties)	PATH			representation representation.items[i] -> representation_item => measure_representation_item
DEFECT_PREDICTION	representation	43		{representation representation.name = 'defect prediction'}
description	descriptive_representation_item.-	45		representation representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
type_of_defect	representation_item.name	43		representation representation.items[i] -> {representation_item => descriptive_representation_item} representation_item representation_item.name
defect_prediction to axis_placement (as location)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'defect location'} representation_item => geometric_representation_item => placement
defect_prediction to shape_aspect (as defect_shape)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
INTEGRATION_INTERVAL	representation	43		{representation representation.name = 'integration interval'}
integration_interval to measure_with_unit (as delta_time)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'delta time'} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
integration_interval to measure_with_unit (as length_of_elapsed_time_interval)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'elapsed time interval'} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
MESHING_CONDITION	representation	43		{representation representation.name = 'meshing condition'}
number_of_elements	measure_with_unit.value_component	41		representation representation.items[i] -> {representation_item representation_item.name = 'number of elements'} representation_item => measure_representation_item <= measure_with_unit measure_with_unit.value_component {measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
number_of_nodes	measure_with_unit.value_component	41		representation representation.items[i] -> {representation_item representation_item.name = 'number of nodes'} representation_item => measure_representation_item <= measure_with_unit

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
				measure_with_unit.value_component {measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
meshing_condition to measure_range_with_unit (as density_of_elements)	PATH			representation representation.items[i] -> {[representation_item representation_item.name = 'density of elements'] [representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier (type_qualifier.name = 'nominal amount') (type_qualifier.name = 'lower limit') (type_qualifier.name = 'upper limit')]} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component {measure_with_unit.unit_component -> unit unit = derived_unit derived_unit}]
meshing_condition to measure_range_with_unit (as density_of_nodes)	PATH			representation representation.items[i] -> {[representation_item representation_item.name = 'density of nodes'] [representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier (type_qualifier.name = 'nominal amount') (type_qualifier.name = 'lower limit') (type_qualifier.name = 'upper limit')]} representation_item =>

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component {measure_with_unit.unit_component -> unit unit = derived_unit derived_unit}]
meshing_condition to shape_aspect (as areas_to_be_meshed)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship shape_aspect_relationship.name = ‘meshed area members’} shape_aspect
SIMULATED_PROPERTY	tensor_representation_item	223		tensor_representation_item <= representation_item
name	representation_item.name	43		tensor_representation_item <= representation_item representation_item.name
simulated_property to scalar_measure (as value)	PATH			tensor_representation_item tensor_representation_item.tensor_value -> tensor_type tensor_type = scalar scalar
simulated_property to tensor_measure (as value)	PATH			tensor_representation_item tensor_representation_item.tensor_value -> tensor_type (tensor_type = symmetric_tensor2_2d symmetric_tensor2_2d) (tensor_type = symmetric_tensor2_3d symmetric_tensor2_3d)

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
				(tensor_type = anisotropic_symmetric_tensor2_2d anisotropic_symmetric_tensor2_2d) (tensor_type = anisotropic_symmetric_tensor2_3d anisotropic_symmetric_tensor2_3d) (tensor_type = isotropic_symmetric_tensor2_3d isotropic_symmetric_tensor2_3d) (tensor_type = orthotropic_symmetric_tensor2_3d orthotropic_symmetric_tensor2_3d)
simulated_property to vector_measure (as value)	PATH			tensor_representation_item tensor_representation_item.tensor_value -> tensor_type tensor_type = tensor1 tensor1
SIMULATION_INPUT	simulation_input	223		simulation_input <= property_definition
simulation_input to computer_file (as setup_file)	PATH			simulation_input document_assigned_item = simulation_input document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type.product_data_type = 'computer file'}
simulation_input to condition_or_assumption (as assumptions)	PATH			simulation_input <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'assumption'} representation

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
simulation_input to condition_or_assumption (as conditions)	PATH			simulation_input <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'condition'} representation
simulation_input to integration_interval (as integration_time_specifications)	PATH			simulation_input <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'integration interval'} representation
simulation_input to property_relationship (as input_parameters)	PATH			simulation_input <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => data_curve_representation
simulation_input to property_value (as input_parameters)	PATH			simulation_input <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => measure_representation_item

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
simulation_input to simulation_input_region (as regions)	PATH			simulation_input <= property_definition <- property_definition_relationship.relating_property_-definition property_definition_relationship property_definition_relationship.related_property_-definition -> property_definition => simulation_input_region
SIMULATION_INPUT_REGION	simulation_input_region	223		simulation_input_region <= property_definition
region_type	descriptive_representation_item.-	45		simulation_input_region <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'region type'} representation_item => descriptive_representation_item descriptive_representation_item.description
simulation_input_region to boundary_condition (as boundary_conditions)	PATH			simulation_input_region <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'boundary condition'} representation
simulation_input_region to item_definition (as associated_items)	PATHsee note 2			simulation_input_region <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
				(product_definition) (product_definition => casting_part) (product_definition => casting_mold_design_element) (product_definition => die_mold) (product_definition => investment_mold) (product_definition => investment_mold_pattern) (product_definition => investment_mold_sprue) (product_definition => sand_mold)
simulation_input_region to meshing_condition (as meshing_conditions)	PATH			simulation_input_region <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'meshing condition'} representation
simulation_input_region to property_relationship (as region_properties)	PATH			simulation_input_region <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => data_curve_representation
simulation_input_region to property_value (as region_properties)	PATH			simulation_input_region <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] ->

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation_item => measure_representation_item
simulation_input_region to shape_representation (as simulation_region_shape)	PATH			simulation_input_region <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'region shape'} representation => shape_representation => (advanced_brep_shape_representation) (faceted_brep_shape_representation) (elementary_brep_shape_representation) (geometrically_bounded_2d_shape_representation)
SIMULATION_OUTPUT	representation	43		{representation representation.name = 'simulation output'}
simulation_output to defect_prediction (as predicted_defects)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'defect prediction'} representation
simulation_output to simulation_result (as results)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'simulation result'} representation
SIMULATION_OUTPUT_REGION	simulation_output_region	223		simulation_output_region <= property_definition
simulation_output_region to shape_representation (as region_shape)	PATH			simulation_output_region <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'region shape'}

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation => shape_representation => (advanced_brep_shape_representation) (faceted_brep_shape_representation) (elementary_brep_shape_representation) (geometrically_bounded_2d_shape_representation)
simulation_output_region to simulation_input_region (as belongs_to)	PATH			simulation_output_region <= property_definition <- property_definition_relationship.related_property_-definition property_definition_relationship property_definition_relationship.relating_property_-definition -> property_definition => simulation_input_region
simulation_output_region to simulation_unit_state (as histories)	PATH			simulation_output_region <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => simulation_unit_state
SIMULATION_RESULT	representation	43		{representation representation.name = 'simulation result'}
simulation_result to measure_with_unit (as begin_time)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'begin time'} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.unit_component] [measure_with_unit.value_component {measure_with_unit.value_component -> measure_value measure_value = time_measure time_measure}]

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
simulation_result to measure_with_unit (as end_time)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'end time'} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.unit_component] [measure_with_unit.value_component {measure_with_unit.value_component -> measure_value measure_value = time_measure time_measure}]
simulation_result to picture (as illustrations)	PATH			representation document_assigned_item = representation document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type.product_data_type = 'picture'}
simulation_result to simulation_output_region (as regions)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition => simulation_output_region
SIMULATION_RUN	simulation_run	223		simulation_run <= action
description	action.description	41		simulation_run <= action action.description
identification	action.name	41		simulation_run <= action action.name

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
simulation_run to process (as associated_process)	PATH			simulation_run <= action action.chosen_method -> action_method
simulation_run to simulation_input (as input)	PATH			simulation_run <= action <- action_assignment.assigned_action action_assignment => casting_action_assignment casting_action_assignment.items[i] -> action_assigned_item action_assigned_item = simulation_input simulation_input
simulation_run to simulation_output (as output)	PATH			simulation_run <= action characterized_action_definition = action characterized_action_definition <- action_property.definition {action_property action_property.name = 'simulation output'} action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation {representation representation.name = 'simulation output'}
simulation_run to simulation_software (as software_used)	PATH			simulation_run <= action supported_item = action supported_item <- action_resource.usage[i] action_resource {action_resource action_resource.kind -> action_resource_type action_resource_type.name = 'simulation software'}

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
SIMULATION_RUN_ RELATIONSHIP	action_relationship	41		
description	action_relationship.description	41		
simulation_run_relationship to simulation_run (as related_run)	PATH			action_relationship action_relationship.related_action -> action => simulation_run
simulation_run_relationship to simulation_run (as relating_run)	PATH			action_relationship action_relationship.relating_action -> action => simulation_run
SIMULATION_SOFTWARE	action_resource	41		{action_resource action_resource.kind -> action_resource_type action_resource_type.name = 'simulation software'}
software_name	action_resource.name	41		
software_version	action_resource.description	41		
simulation_software to date (as release_date)	PATH			action_resource date_assigned_item = action_resource date_assigned_item <- casting_date_assignment.items[i] casting_date_assignment <= date_assignment {date_assignment date_assignment.role -> date_role date_role.name = 'software release date'} date_assignment.assigned_date -> date => calendar_date
SIMULATION_UNIT	property_definition	41		{property_definition property_definition.name = 'simulation state'}
name	descriptive_representation_ item.-	45		property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item

Table 10 – Continued

Application element	AIM element	Source	Rules	Reference path
				representation_item.name = 'simulation unit name'} representation_item => descriptive_representation_item descriptive_representation_item.description
simulation_unit to axis_placement (as unit_location)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'unit location'} representation_item => geometric_representation_item => placement
simulation_unit to shape_aspect (as unit_shape)	PATH			property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect
simulation_unit to simulation_output_region (as belongs_to)	PATH			property_definition <- property_definition_relationship.related_property_- definition {property_definition_relationship property_definition_relationship.name = 'output region unit'} property_definition_relationship property_definition_relationship.relating_property_- definition -> property_definition => simulation_output_region
simulation_unit to simulation_unit_state (as histories)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => simulation_unit_state

Table 10 – Concluded

Application element	AIM element	Source	Rules	Reference path
SIMULATION_UNIT_STATE	simulation_unit_state	223		simulation_unit_state <= representation
simulation_unit_state to measure_with_unit (as evaluation_time)	PATH			simulation_unit_state <= representation representation.items[i] -> {representation_item representation_item.name = 'evaluation time'} representation_item => measure_representation_item <= {measure_with_unit => time_measure_with_unit} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
simulation_unit_state to simulated_property (as properties)	PATH			simulation_unit_state <= representation representation.items[i] -> representation_item

Table 11 – Mapping table for substance_composition UoF

Application element	AIM element	Source	Rules	Reference path
GEOMETRY_ CHARACTERIZATION	product_definition_shape	41		
description	property_definition.description	41		product_definition_shape <= property_definition property_definition.description
geometry_characterization to axis_placement (as orientation)	PATH			product_definition_shape <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation} representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement
geometry_characterization to measure_range_with_unit (as characteristic_size)	PATH			product_definition_shape <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation} representation representation.items[i] -> {representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier

Table 11 – Continued

Application element	AIM element	Source	Rules	Reference path
				type_qualifier (type_qualifier.name = 'nominal amount') (type_qualifier.name = 'lower limit') (type_qualifier.name = 'upper limit')} representation_item => measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
geometry_characterization to shape_representation (as shape)	PATH			product_definition_shape <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => shape_representation => (advanced_brep_shape_representation) (faceted_brep_shape_representation) (elementary_brep_shape_representation) (geometrically_bounded_2d_shape_representation)
SUBSTANCE	product_definition	41		{product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'material'}
description	product_definition.description	41		
physical_form	property_definition.name	41		product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition

Table 11 – Continued

Application element	AIM element	Source	Rules	Reference path
				property_definition.name
substance to substance_composition_element (as composition)	PATH			product_definition <- product_definition_relationship.relating_product_- definition product_definition_relationship {product_definition_relationship => product_material_composition_relationship} product_definition_relationship.related_product_- definition -> product_definition => chemical_structure_element
substance to substance_structure (as structure)	PATH			product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition => material_property
SUBSTANCE_ COMPOSITION_ELEMENT	chemical_structure_element	223		chemical_structure_element <= product_definition
element	product.name	41		chemical_structure_element <= product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.name {(product.name = 'Actinium') (product.name = 'Aluminum') (product.name = 'Americium') (product.name = 'Antimony') (product.name = 'Argon') (product.name = 'Arsenic') (product.name = 'Astatine')

Table 11 – Continued

Application element	AIM element	Source	Rules	Reference path
				(product.name = ‘Barium’)
				(product.name = ‘Berkelium’)
				(product.name = ‘Beryllium’)
				(product.name = ‘Bismuth’)
				(product.name = ‘Boron’)
				(product.name = ‘Bromine’)
				(product.name = ‘Cadmium’)
				(product.name = ‘Calcium’)
				(product.name = ‘Californium’)
				(product.name = ‘Carbon’)
				(product.name = ‘Cerium’)
				(product.name = ‘Cesium’)
				(product.name = ‘Chlorine’)
				(product.name = ‘Chromium’)
				(product.name = ‘Cobalt’)
				(product.name = ‘Copper’)
				(product.name = ‘Curium’)
				(product.name = ‘Dysprosium’)
				(product.name = ‘Einsteinium’)
				(product.name = ‘Erbium’)
				(product.name = ‘Europium’)
				(product.name = ‘Fermium’)
				(product.name = ‘Fluorine’)
				(product.name = ‘Francium’)
				(product.name = ‘Gadolinium’)
				(product.name = ‘Gallium’)
				(product.name = ‘Germanium’)
				(product.name = ‘Gold’)
				(product.name = ‘Hafnium’)
				(product.name = ‘Helium’)
				(product.name = ‘Holmium’)
				(product.name = ‘Hydrogen’)
				(product.name = ‘Indium’)
				(product.name = ‘Iodine’)
				(product.name = ‘Iridium’)
				(product.name = ‘Iron’)
				(product.name = ‘Krypton’)
				(product.name = ‘Lanthanum’)
				(product.name = ‘Lawrencium’)
				(product.name = ‘Lead’)
				(product.name = ‘Lithium’)
				(product.name = ‘Lutetium’)

Table 11 – Continued

Application element	AIM element	Source	Rules	Reference path
				(product.name = 'Magnesium') (product.name = 'Manganese') (product.name = 'Mendelevium') (product.name = 'Mercury') (product.name = 'Molybdenum') (product.name = 'Neodymium') (product.name = 'Neon') (product.name = 'Neptunium') (product.name = 'Nickel') (product.name = 'Niobium') (product.name = 'Nitrogen') (product.name = 'Nobelium') (product.name = 'Osmium') (product.name = 'Oxygen') (product.name = 'Palladium') (product.name = 'Phosphorus') (product.name = 'Platinum') (product.name = 'Plutonium') (product.name = 'Polonium') (product.name = 'Potassium') (product.name = 'Praseodymium') (product.name = 'Promethium') (product.name = 'Protactinium') (product.name = 'Radium') (product.name = 'Radon') (product.name = 'Rhenium') (product.name = 'Rhodium') (product.name = 'Rubidium') (product.name = 'Ruthenium') (product.name = 'Samarium') (product.name = 'Scandium') (product.name = 'Selenium') (product.name = 'Silicon') (product.name = 'Silver') (product.name = 'Sodium') (product.name = 'Strontium') (product.name = 'Sulfur') (product.name = 'Tantalum') (product.name = 'Technetium') (product.name = 'Tellurium') (product.name = 'Terbium') (product.name = 'Thallium')

Table 11 – Continued

Application element	AIM element	Source	Rules	Reference path
				(product.name = 'Thorium') (product.name = 'Thulium') (product.name = 'Tin') (product.name = 'Titanium') (product.name = 'Uranium') (product.name = 'Vanadium') (product.name = 'Wolfram') (product.name = 'Xenon') (product.name = 'Ytterbium') (product.name = 'Yttrium') (product.name = 'Zinc') (product.name = 'Zirconium')}
substance_composition_element to measure_range_with_unit (as amount)	PATH			chemical_structure_element <= product_definition <- product_definition_relationship.related_product_- definition product_definition_relationship => product_material_composition_relationship product_material_composition_- relationship.constituent_amount[i] -> {measure_with_unit <- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier (type_qualifier.name = 'nominal amount') (type_qualifier.name = 'lower limit') (type_qualifier.name = 'upper limit')} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
SUBSTANCE_STRUCTURE	material_property	45		
description	property_definition.description	41		material_property <= property_definition property_definition.description

Table 11 – Continued

Application element	AIM element	Source	Rules	Reference path
type_of_structure	property_definition.name	41		material_property <= property_definition property_definition.name
substance_structure to picture (as illustration)	PATH			material_property document_assigned_item = material_property document_assigned_item <- casting_document_reference.items[i] casting_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type.product_data_type = ‘picture’}
substance_structure to specification_reference (as reference_to)	PATH			material_property document_assigned_item = material_property document_assigned_item <- casting_document_reference.items[i] casting_document_reference {casting_document_reference <= document_reference document_reference.assigned_document -> document document.kind -> document_type document_type.product_data_type = ‘specification’}
substance_structure to substance_structure_element (as elements)	PATH			material_property <= property_definition <- property_definition_relationship.relating_property_- definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = ‘property structure element’} property_definition_relationship.related_property_- definition ->

Table 11 – Continued

Application element	AIM element	Source	Rules	Reference path
				{property_definition => material_property} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition characterized_product_definition = product_definition product_definition => physical_structure_element
SUBSTANCE_STRUCTURE_ELEMENT	physical_structure_element	223		physical_structure_element <= product_definition
type_of_element	product_definition.description	41		physical_structure_element <= product_definition product_definition.description
substance_structure_element to geometry_characterization (as characteristic_geometry)	PATH			physical_structure_element <= product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape
substance_structure_element to measure_range_with_unit (as amount)	PATH			physical_structure_element <= product_definition <- product_definition_relationship.related_product_- definition product_definition_relationship => product_material_composition_relationship product_material_composition_- relationship.constituent_amount -> {[measure_with_unit => ratio_measure_with_unit] [measure_with_unit <- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier

Table 11 – Concluded

Application element	AIM element	Source	Rules	Reference path
				type_qualifier (type_qualifier.name = 'nominal amount') (type_qualifier.name = 'lower limit') (type_qualifier.name = 'upper limit')]} measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
substance_structure_element to substance (as made_of)	PATH			physical_structure_element <= product_definition <- product_definition_relationship.related_product_-definition product_definition_relationship {[product_definition_relationship => product_material_composition_relationship] [product_definition_relationship product_definition_relationship.name = 'structure element']} product_definition_relationship.relating_product_-definition -> product_definition {product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category {product_category.name = 'material'}

Table 12 – Mapping table for tolerance UoF

Application element	AIM element	Source	Rules	Reference path
DATUM_REFERENCE_FRAME	(datum)	47		
	([datum]	47		
	[(datum_feature)	47		
	(datum_target_feature)]	47		
DIMENSIONAL_TOLERANCE	(dimensional_size)	47		
	(dimensional_location)	47		
GEOMETRIC_TOLERANCE	(geometric_tolerance)	47		
	(geometric_tolerance_with_datum_reference)	47		
	(geometric_tolerance_with_defined_unit)	47		
	(angularity_tolerance)	223		
	(circular_runout_tolerance)	223		
	(concentricity_tolerance)	223		
	(parallelism_tolerance)	223		
	(perpendicularity_tolerance)	223		
	(position_tolerance)	223		
	(symmetry_tolerance)	223		

Table 12 – Continued

Application element	AIM element	Source	Rules	Reference path
	(total_runout_tolerance)	223		
	(linear_profile_tolerance)	223		
	(surface_profile_tolerance)	223	11	(angularity_tolerance <=) (circular_runout_tolerance <=) (concentricity_tolerance <=) (parallelism_tolerance <=) (perpendicularity_tolerance <=) (position_tolerance <=) (symmetry_tolerance <=) (total_runout_tolerance <=) geometric_tolerance_with_datum_reference (linear_profile_tolerance <=) (surface_profile_tolerance <=) geometric_tolerance
geometric_tolerance to datum_reference_frame (as with_respect_to)	PATH			(angularity_tolerance <= geometric_tolerance_with_datum_reference) (circular_runout_tolerance <= geometric_tolerance_with_datum_reference) (concentricity_tolerance <= geometric_tolerance_with_datum_reference) (parallelism_tolerance <= geometric_tolerance_with_datum_reference) (perpendicularity_tolerance <= geometric_tolerance_with_datum_reference) (position_tolerance <= geometric_tolerance_with_datum_reference) (symmetry_tolerance <= geometric_tolerance_with_datum_reference) (total_runout_tolerance <= geometric_tolerance_with_datum_reference) (geometric_tolerance_with_datum_reference) geometric_tolerance_with_datum_reference.datum_- system[i] -> datum_reference datum_reference.referenced_datum -> (datum)

Table 12 – Continued

Application element	AIM element	Source	Rules	Reference path
				([datum] [(datum_feature) (datum_target_feature)]
MISMATCH_TOLERANCE	geometric_tolerance	47		{geometric_tolerance geometric_tolerance.name = 'mismatch'}
mismatch_tolerance to measure_with_unit (as maximum_amount)	PATH			geometric_tolerance geometric_tolerance.magnitude -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
mismatch_tolerance to parting_surface (as with_respect_to)	PATH			geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'parting surface'}
TOLERANCE	(dimensional_size)	47		
	(dimensional_location)	47		
	(geometric_tolerance)	47		
	(geometric_tolerance_with_datum_reference)	47		
	(geometric_tolerance_with_defined_unit)	47		
	(angularity_tolerance)	223		
	(circular_runout_tolerance)	223		
	(concentricity_tolerance)	223		
	(parallelism_tolerance)	223		
	(perpendicularity_tolerance)	223		

Table 12 – Continued

Application element	AIM element	Source	Rules	Reference path
	(position_tolerance)	223		
	(symmetry_tolerance)	223		
	(total_runout_tolerance)	223		
	(linear_profile_tolerance)	223		
	(surface_profile_tolerance)	223	11	(angularity_tolerance <=) (circular_runout_tolerance <=) (concentricity_tolerance <=) (parallelism_tolerance <=) (perpendicularity_tolerance <=) (position_tolerance <=) (symmetry_tolerance <=) (total_runout_tolerance <=) geometric_tolerance_with_datum_reference (linear_profile_tolerance <=) (surface_profile_tolerance <=) geometric_tolerance
name	dimensional_size.name	47		
	shape_aspect_relationship.name	41		
	geometric_tolerance.name	47		(dimensional_location <= shape_aspect_relationship shape_aspect_relationship.name) (angularity_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'angularity'}) (circular_runout_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'circular runout'}) (concentricity_tolerance <=

Table 12 – Continued

Application element	AIM element	Source	Rules	Reference path
				geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'concentricity'}) (linear_profile_tolerance <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'linear profile'}) (parallelism_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'parallelism'}) (perpendicularity_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'perpendicularity'}) (position_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'position'}) (surface_profile_tolerance <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'surface profile'}) (symmetry_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'symmetry'}) (total_runout_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name {geometric_tolerance.name = 'total runout'}) (geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name) (geometric_tolerance_with_defined_unit <= geometric_tolerance

Table 12 – Concluded

Application element	AIM element	Source	Rules	Reference path
				geometric_tolerance.name {(geometric_tolerance.name = 'flatness') (geometric_tolerance.name = 'straightness')}) (geometric_tolerance geometric_tolerance.name {(geometric_tolerance.name) (geometric_tolerance.name = 'mismatch') (geometric_tolerance.name = 'circularity') (geometric_tolerance.name = 'cylindricity') (geometric_tolerance.name = 'flatness') (geometric_tolerance.name = 'straightness')}
tolerance to casting_design_feature (as specify_allowance_on)	PATH			

The following rules are referenced in the proceeding tables:

1) approval_requires_approval_date_time

2) approval_requires_approval_person_organization

3) approvals_are_assigned

5.2 AIM EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources and contains the types, entity specializations, rules, and functions that are specific to this Part of ISO 10303.

EXPRESS specification:

```
*)
SCHEMA metal_casting;

  USE FROM aic_advanced_brep;                   -- ISO 10303-514

  USE FROM aic_elementary_brep;                 -- ISO 10303-513

  USE FROM aic_faceted_brep;                    -- ISO 10303-512

  USE FROM aic_geometrically_bounded_2d_wireframe;-- ISO 10303-503

  USE FROM action_schema                         -- ISO 10303-41
      (action,
       action_method,
       action_relationship,
       action_request_solution,
       action_resource,
       directed_action,
       versioned_action_request);

  USE FROM application_context_schema            -- ISO 10303-41
      (application_context,
       application_protocol_definition);

  USE FROM approval_schema                       -- ISO 10303-41
      (approval,
       approval_date_time,
       approval_person_organization);

  USE FROM date_time_schema                      -- ISO 10303-41
      (calendar_date,
       date_and_time);

  USE FROM document_schema                       -- ISO 10303-41
      (document,
       document_usage_constraint);

  USE FROM effectivity_schema                    -- ISO 10303-41
```

```
    (effectivity,
     lot_effectivity);

USE FROM fea_scalar_vector_tensor_schema
    (scalar,
     tensor1,
     tensor1_2d,
     tensor1_3d,
     symmetric_tensor2_2d,
     symmetric_tensor2_3d,
     anisotropic_symmetric_tensor2_2d,
     anisotropic_symmetric_tensor2_3d,
     isotropic_symmetric_tensor2_3d,
     orthotropic_symmetric_tensor2_3d);

USE FROM management_resources_schema            -- ISO 10303-41
    (action_assignment,
     action_request_assignment,
     approval_assignment,
     date_and_time_assignment,
     date_assignment,
     document_reference,
     effectivity_assignment,
     organization_assignment,
     person_and_organization_assignment);

USE FROM material_property_definition_schema    -- ISO 10303-45
    (material_property,
     product_material_composition_relationship,
     property_definition_relationship);

USE FROM measure_schema                         -- ISO 10303-41
    (count_measure,
     derived_unit,
     length_measure_with_unit,
     length_unit,
     measure_with_unit,
     plane_angle_measure_with_unit,
     plane_angle_unit,
     ratio_measure_with_unit,
     ratio_unit,
     time_measure);

USE FROM method_definition_schema               -- ISO 10303-49
    (concurrent_action_method,
     sequential_method);

USE FROM process_property_schema                -- ISO 10303-49
    (action_resource_requirement,
     process_product_association,
     product_definition_process,
     property_process);
```

```
USE FROM process_property_representation_schema -- ISO 10303-49
    (action_property_representation,
     resource_property_representation);

USE FROM product_definition_schema             -- ISO 10303-41
    (product_category_relationship,
     product_definition,
     product_definition_formation,
     product_related_product_category);

USE FROM product_property_definition_schema     -- ISO 10303-41
    (characterized_object,
     property_definition,
     shape_aspect);

USE FROM product_property_representation_schema -- ISO 10303-41
    (shape_definition_representation);

USE FROM product_structure_schema              -- ISO 10303-44
    (assembly_component_usage,
     make_from_usage_option);

USE FROM qualified_measure_schema              -- ISO 10303-45
    (descriptive_representation_item,
     measure_qualification,
     measure_representation_item,
     qualified_representation_item,
     type_qualifier);

USE FROM representation_schema                 -- ISO 10303-43
    (representation,
     representation_item,
     representation_relationship);

USE FROM shape_dimension_schema                -- ISO 10303-47
    (dimensional_location,
     dimensional_size);

USE FROM shape_tolerance_schema                -- ISO 10303-47
    (geometric_tolerance,
     geometric_tolerance_with_datum_reference,
     geometric_tolerance_with_defined_unit);
(*
```

NOTE – The schemas referenced above can be found in the following parts of ISO 10303:

aic_advanced_brep ISO 10303-514

aic_elementary_brep ISO 10303-513

aic_faceted_brep ISO 10303-512

aic_geometrically_bounded_2d_wireframe ISO 10303-503

action_schema ISO 10303-41

application_context_schema ISO 10303-41

approval_schema ISO 10303-41

date_time_schema ISO 10303-41

document_schema ISO 10303-41

effectivity_schema ISO 10303-41

fea_scalar_vector_tensor_schema ISO 10303-104

management_resources_schema ISO 10303-41

material_property_definition_schema ISO 10303-45

measure_schema ISO 10303-41

method_definition_schema ISO 10303-49

person_organization_schema ISO 10303-41

process_property ISO 10303-49

process_property_representation_schema ISO 10303-49

product_definition_schema ISO 10303-41

product_property_definition_schema ISO 10303-41

product_property_representation_schema ISO 10303-41

product_structure_schema ISO 10303-44

qualified_measure_schema ISO 10303-45

representation_schema ISO 10303-43

shape_aspect_definition_schema ISO 10303-47

shape_dimension_schema ISO 10303-47

shape_tolerance_schema ISO 10303-47

5.2.1 Exchange of design and manufacturing product information for cast parts type definitions

5.2.1.1 action_assigned_item

An action_assigned_item identifies a casting_part_definition, characterized_object, product_definition, property_- definition, shape_aspect, or simulation_input to which an action may be assigned.

EXPRESS specification:

```
*)
TYPE action_assigned_item = SELECT
  (casting_part_definition,
   characterized_object,
   product_definition,
   property_definition,
   shape_aspect,
   simulation_input);
END_TYPE;
(*
```

5.2.1.2 action_request_assigned_item

An action_request_assigned_item identifies a product_definition, product_definition_formation, property_- definition, or shape_aspect to which a versioned_action_request may be assigned.

EXPRESS specification:

```
*)
TYPE action_request_assigned_item = SELECT
  (product_definition,
   product_definition_formation,
   property_definition,
   shape_aspect);
END_TYPE;
(*
```

5.2.1.3 approval_assigned_item

An approval_assigned_item identifies a product_definition_formation or process_plan_version to which an approval may be assigned.

EXPRESS specification:

```
*)
```

```
TYPE approval_assigned_item = SELECT
  (product_definition_formation,
   process_plan_version);
END_TYPE;
(*
```

5.2.1.4 change_item

A change_item identifies a characterized_object to which a change may be assigned.

EXPRESS specification:

```
*)
TYPE change_item = SELECT
  (characterized_object);
END_TYPE;
(*
```

5.2.1.5 date_and_time_assigned_item

A date_and_time_assigned_item identifies a casting_part_definition, directed_action, heat_effectivity, lot_- effectivity, process_plan_version, representation, or versioned_action_request to which a date_-and_time may be assigned.

EXPRESS specification:

```
*)
TYPE date_and_time_assigned_item = SELECT
  (casting_part_definition,
   directed_action,
   heat_effectivity,
   lot_effectivity,
   process_plan_version,
   representation,
   versioned_action_request);
END_TYPE;
(*
```

5.2.1.6 date_assigned_item

A date_assigned_item identifies a action_resource, casting_part_definition, directed_action, heat_-effectivity, lot_effectivity, process_plan_version, representation, or versioned_action_request to which a date may be assigned.

EXPRESS specification:

```
*)
TYPE date_assigned_item = SELECT
  (action_resource,
   casting_part_definition,
   directed_action,
   heat_effectivity,
   lot_effectivity,
   process_plan_version,
   representation,
   versioned_action_request);
END_TYPE;
(*
```

5.2.1.7 document_assigned_item

A document_assigned_item identifies a action_method, action_resource, action_resource_requirement, dimensional_- location, dimensional_size, geometric_tolerance, identification_marking, material_property, product_definition_- process, property_process, property_definition, representation, or simulation_- input to which a document may be assigned.

EXPRESS specification:

```
*)
TYPE document_assigned_item = SELECT
  (action_method,
   action_resource,
   action_resource_requirement,
   dimensional_location,
   dimensional_size,
   geometric_tolerance,
   identification_marking,
   material_property,
   product_definition_process,
   property_process,
   property_definition,
   representation,
   simulation_input);
END_TYPE;
(*
```

5.2.1.8 effectivity_assigned_item

An effectivity_assigned_item identifies a casting_part_definition to which an effectivity may be assigned.

EXPRESS specification:

```
*)
```

```
TYPE effectivity_assigned_item = SELECT
  (casting_part_definition);
END_TYPE;
(*
```

5.2.1.9 organization_assigned_item

An organization_assigned_item identifies a document to which an organization may be assigned.

EXPRESS specification:

```
*)
TYPE organization_assigned_item = SELECT
  (document);
END_TYPE;
(*
```

5.2.1.10 person_and_organization_assigned_item

A person_and_organization_assigned_item identifies a characterized_object, directed_action, process_plan_version, product_definition, or versioned_action_request to which a person_and_organization may be assigned.

EXPRESS specification:

```
*)
TYPE person_and_organization_assigned_item = SELECT
  (characterized_object,
   directed_action,
   process_plan_version,
   product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.1.11 process_plan_input_item

A process_plan_input_item identifies a characterized_object or product_definition which is the input to a process plan.

EXPRESS specification:

```
*)
TYPE process_plan_input_item = SELECT
  (characterized_object,
```

```
   product_definition);
END_TYPE;
(*
```

5.2.1.12 tensor_type

A tensor_type is a scalar, tensor1, symmetric_tensor2_2d, symmetric_tensor2_3d, anisotropic_symmetric_tensor2_2d, anisotropic_symmetric_tensor2_3d, isotropic_symmetric_tensor2_3d, or orthotropic_symmetric_tensor2_3d that gives the value for a tensor_representation_item.

EXPRESS specification:

```
*)
TYPE tensor_type = SELECT
  (scalar,
   tensor1,
   symmetric_tensor2_2d,
   symmetric_tensor2_3d,
   anisotropic_symmetric_tensor2_2d,
   anisotropic_symmetric_tensor2_3d,
   isotropic_symmetric_tensor2_3d,
   orthotropic_symmetric_tensor2_3d);
END_TYPE;
(*
```

5.2.2 Exchange of design and manufacturing product information for cast parts entities

5.2.2.1 Exchange of design and manufacturing product information for cast parts entity definitions

5.2.2.1.1 angularity_tolerance

An angularity_tolerance is a type of geometric_tolerance_with_datum_reference which represents the maximum variation of a considered shape_aspect.

EXPRESS specification:

```
*)
ENTITY angularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'angularity';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system)
```

```
      = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the angularity_tolerance shall be 'angularity'.

WR2: There shall be exactly one datum_reference in the datum_system set for the angularity_tolerance.

5.2.2.1.2 casting_action_assignment

A casting_action_assignment relates an action to a piece of product data.

EXPRESS specification:

```
*)
ENTITY casting_action_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF action_assigned_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of casting_part_definition, characterized_object, property_definition, product_definition, shape_aspect, or simulation_input to which the action is being related.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the casting_action_assignment entity:

- design_update_requires_item_definition (See **??**).

5.2.2.1.3 casting_action_request_assignment

A casting_action_request_assignment relates a versioned_action_request to a piece of product data.

EXPRESS specification:

```
*)
ENTITY casting_action_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF action_request_assigned_item;
```

```
END_ENTITY;
(*
```

Attribute definitions:

items: the set of product_definition, product_definition_formation, property_definition, or shape_aspect to which the versioned_action_request is being related.

5.2.2.1.4 casting_approval_assignment

A casting_approval_assignment specifies those items for which an approval is established.

EXPRESS specification:

```
*)
ENTITY casting_approval_assignment
  SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF approval_assigned_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of process_plan_version or product_definition_formation for which an approval may be established.

5.2.2.1.5 casting_date_and_time_assignment

A casting_date_and_time_assignment relates a date_and_time to a piece of product data.

EXPRESS specification:

```
*)
ENTITY casting_date_and_time_assignment
  SUBTYPE OF (date_and_time_assignment);
  items : SET [1:?] OF date_and_time_assigned_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of casting_part_definition, directed_action, heat_effectivity, lot_effectivity, process_plan_version, representation, or versioned_action_request to which the date_and_time is being related.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the casting_date_and_time_assignment entity:

- design_request_requires_date_or_date_and_time (See ??);
- design_update_requires_date_or_date_and_time (See ??);
- lot_requires_date_or_date_and_time (See ??).

5.2.2.1.6 casting_date_assignment

A casting_date_assignment relates a date to a piece of product data.

EXPRESS specification:

```
*)
ENTITY casting_date_assignment
  SUBTYPE OF (date_assignment);
  items : SET [1:?] of date_assigned_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of action_resource, casting_part_definition, directed_action, heat_effectivity, lot_effectivity, process_plan_version, representation, or versioned_action_request to which the date is being related.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the casting_date_assignment entity:

- design_request_requires_date_or_date_and_time (See ??);
- design_update_requires_date_or_date_and_time (See ??);
- lot_requires_date_or_date_and_time (See ??).

5.2.2.1.7 casting_document_reference

A casting_document_reference relates a document to a piece of product data.

EXPRESS specification:

```
*)
ENTITY casting_document_reference
```

```
  SUBTYPE OF (document_reference);
  items : SET [1:?] OF document_assigned_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of action_method, action_resource, action_resource_requirement, dimesional_location, dimensional_size, geometrical_tolerance, identification_marking, material_property, property_definition, representation, or simulation_input to which the document is being related.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the casting_document_reference entity:

- item_requirement_requires_specification (See **??**).

5.2.2.1.8 casting_effectivity_assignment

A casting_effectivity_assignment relates an effectivity to a piece of product data.

EXPRESS specification:

```
*)
ENTITY casting_effectivity_assignment
  SUBTYPE OF (effectivity_assignment);
  items : SET [1:?] OF effectivity_assigned_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of casting_part_definition to which the effectivity is being related.

5.2.2.1.9 casting_machining_allowance

A casting_machining_allowance is

EXPRESS specification:

```
*)
ENTITY casting_machining_allowance
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: SELF.description IN ['explicit surface allowance',
```

```
         'allowance along vector', 'allowance along normal'];
    WR2: SIZEOF (QUERY (sar <* USEDIN (SELF,
         'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
         'RELATING_SHAPE_ASPECT') |
         sar.name = 'geometry modification')) = 1;
    WR3: SIZEOF (QUERY (pd <*
         USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
         NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         pdr.used_representation.name = 'resultant geometry')) = 1))) = 0;
    WR4: (NOT (SELF.name = 'explicit surface allowance')) OR
         (SIZEOF (QUERY (sar <* USEDIN (SELF,
         'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
         'RELATING_SHAPE_ASPECT') |
         sar.name = 'enveloping geometry')) = 1);
    WR5: (NOT (SELF.name IN ['allowance along vector',
         'allowance along normal'])) OR
         (SIZEOF (QUERY (pd <*
         USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
         NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         pdr.used_representation.name = 'feature parameters')) = 1))) = 0);
    WR6: (NOT (SELF.name = 'allowance along vector')) OR
         (SIZEOF (QUERY (pd <*
         USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
         NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
         'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         pdr.used_representation.name = 'feature parameters') |
         NOT ({2 <= SIZEOF (par_rep.used_representation.items) <= 4}))) = 0)))
         = 0);
    WR7: (NOT (SELF.name = 'allowance along vector')) OR
         (SIZEOF (QUERY (pd <*
         USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
         NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
         'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         pdr.used_representation.name = 'feature parameters') |
         NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
         'METAL_CASTING.VECTOR' IN TYPEOF (it))) = 1))) = 0))) = 0);
    WR8: (NOT (SELF.name = 'allowance along vector')) OR
         (SIZEOF (QUERY (pd <*
         USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
         NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
         'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         pdr.used_representation.name = 'feature parameters') |
         NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
         (SIZEOF (['METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
         'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
         'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF (it)) = 3) AND
         (it.name = 'offset amount') AND
         (SIZEOF (it.qualifiers = 1)) AND
         (SIZEOF (QUERY (qual <* it.qualifiers |
         NOT ('METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF (qual))) = 0)))) <=
```

```
       3))) = 0))) = 0);
  WR9: (NOT (SELF.name = 'allowance along normal')) OR
       (SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
       'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'feature parameters') |
       NOT ({1 <= SIZEOF (par_rep.used_representation.items) <= 3}))) = 1)))
       = 0);
 WR10: (NOT (SELF.name = 'allowance along normal')) OR
       (SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
       'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'feature parameters') |
       NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
       (SIZEOF (['METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
       'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
       'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF (it)) = 3) AND
       (it.name = 'thickness') AND
       (SIZEOF (it.qualifiers = 1)) AND
       (SIZEOF (QUERY (qual <* it.qualifiers |
       NOT ('METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF (qual))) = 0)))) <=
       3))) = 0))) = 0);
END_ENTITY;
(*
```

Formal propositions:

WR1: The casting_machining_allowance shall have a description of explicit surface allowance, allowance along vector, or allowance along normal.

WR2: The casting_machining_allowance shall be the relating_shape_aspect in exactly one shape_aspect_relationship with the name of geometry modification.

WR3: The casting_machining_allowance shall have exactly one representation with a name of resultant geometry.

WR4: If the name of the casting_machining_allowance is explicit surface allowance, the casting_machining_allowance shall be the relating_shape_aspect in exactly one shape_aspect_relationship with the name of enveloping geometry.

WR5: If the name of the casting_machining_allowance is allowance along vector or allowance along normal, the casting_machining_allowance shall have exactly one representation with a name of feature parameters.

WR6: If the name of the casting_machining_allowance is allowance along vector, the casting_machining_allowance shall have between two and four representation_items in the items set of the representation with the name of feature parameters.

WR7: If the name of the casting_machining_allowance is allowance along vector, the casting_machining_allowance shall have exactly one representation_item in the items set of the representation with the name of feature parameters of type vector.

WR8: If the name of the casting_machining_allowance is allowance along vector, the casting_-machining_allowance shall have at most three representation_items with a name of offset amount in the items set of the representation with the name of feature parameters that are a complex instance of type measure_representation_item, length_measure_with_unit, and qualified_representation_item with exactly one qualifier that is a type_qualifier.

WR9: If the name of the casting_machining_allowance is allowance along normal, the casting_-machining_allowance shall have between one and three representation_items in the items set of the representation with the name of feature parameters.

WR10: If the name of the casting_machining_allowance is allowance along normal, the casting_-machining_allowance shall have at most three representation_items with a name of thickness in the items set of the representation with the name of feature parameters that are a complex instance of type measure_representation_item, length_measure_with_unit, and qualified_representation_item with exactly one qualifier that is a type_qualifier.

5.2.2.1.10 casting_mold_design_element

A casting_mold_design_element is a component in the casing into which a substance is poured to make a casting part.

EXPRESS specification:

```
*)
ENTITY casting_mold_design_element
  SUBTYPE OF (product_definition);
WHERE
  WR1: SELF.frame_of_reference.name IN
       ['sand casting', 'die casting', 'investment casting'];
  WR2: SIZEOF (QUERY (acu <* QUERY (pdr <* USEDIN (SELF,
       'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
       'RELATED_PRODUCT_DEFINITION') |
       'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF (pdr)) |
       SIZEOF (['METAL_CASTING.SAND_MOLD_DEFINITION',
       'METAL_CASTING.DIE_MOLD_DEFINITION',
       'METAL_CASTING.INVESTMENT_MOLD_DEFINITION'] * TYPEOF
       (acu.relating_product_definition)) = 1)) >= 1;
  WR3: SELF.frame_of_reference.life_cycle_stage = 'design';
  WR4: SELF.description IN ['die component', 'die sub-assembly',
       'core', 'flask', 'pattern and rigging assembly', 'pattern plate',
       'rigging component', 'insert', 'in-mold rigging component', 'pattern',
       'sand mold component', 'investment pattern', 'sprue component'];
  WR5: (NOT (SELF.description IN ['pattern plate', 'rigging component',
       'pattern'])) OR
       (SIZEOF (QUERY (acu <* QUERY (pdr <* USEDIN (SELF,
       'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
       'RELATED_PRODUCT_DEFINITION') |
       'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF (pdr)) |
```

```
      ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
      TYPEOF (acu.relating_product_definition)) AND
      (acu.relating_product_definition.description =
      'pattern and rigging assembly'))) >= 1);
  WR6: (NOT (SELF.description = 'sand mold component')) OR
      (SIZEOF (QUERY (pdu <* QUERY (pdr <* USEDIN (SELF,
      'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
      'RELATING_PRODUCT_DEFINITION') |
      'METAL_CASTING.PRODUCT_DEFINITION_USAGE' IN TYPEOF (pdr)) |
      ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
      TYPEOF (pdu.relating_product_definition)) AND
      (pdu.relating_product_definition.description =
      'flask'))) = 1);
  WR7: (NOT (SELF.description = 'sand mold component')) OR
      (SIZEOF (QUERY (pdu <* QUERY (pdr <* USEDIN (SELF,
      'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
      'RELATING_PRODUCT_DEFINITION') |
      'METAL_CASTING.PRODUCT_DEFINITION_USAGE' IN TYPEOF (pdr)) |
      ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
      TYPEOF (pdu.relating_product_definition)) AND
      (pdu.relating_product_definition.description =
      'pattern and rigging assembly'))) = 1);
  WR8: (NOT (SELF.description = 'pattern and rigging assembly')) OR
      (SIZEOF (QUERY (pdu <* QUERY (pdr <* USEDIN (SELF,
      'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
      'RELATING_PRODUCT_DEFINITION') |
      'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF (pdr)) |
      ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
      TYPEOF (pdu.relating_product_definition)) AND
      (pdu.relating_product_definition.description =
      'pattern plate'))) = 1);
  WR9: (NOT (SELF.description = 'investment pattern')) OR
      (SIZEOF (QUERY (pdu <* QUERY (pdr <* USEDIN (SELF,
      'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
      'RELATING_PRODUCT_DEFINITION') |
      'METAL_CASTING.PRODUCT_DEFINITION_USAGE' IN TYPEOF (pdr)) |
      ('METAL_CASTING.DIE_MOLD_DEFINITION' IN
      TYPEOF (pdu.related_product_definition)) AND
      (pdu.relating_product_definition.frame_of_reference.name =
      'investment casting'))) = 1);
END_ENTITY;
(*
```

Formal propositions:

WR1: A casting_mold_design_element shall have a frame of reference of sand casting, die casting, or investment casting.

WR2: Each casting_mold_design_element shall be the related_product_definition in an assembly_component_usage that references a die_mold_definition, investment_mold_definition, or sand_mold_definition as the relating_product_definition.

WR3: A casting_mold_design_element shall have a life_cycle_stage of design.

WR4: A casting_mold_design_element shall have a description that specifies its role within the design. The roles of the casting_mold_design_element are 'die component', 'die sub-assembly', 'core', 'flask', 'pattern and rigging assembly', 'pattern plate', 'rigging component', 'insert', 'in-mold rigging component', 'pattern', 'sand mold component', 'investment pattern', or 'sprue component'.

WR5: A casting_mold_design_element with the role of 'pattern plate', 'rigging component', or 'pattern' shall be the related_product_definition in at least one assembly_component_usage with a relating_product_definition that references a casting_mold_design_element with a role of 'pattern and rigging assembly'.

WR6: A casting_mold_design_element with the role of 'sand mold component' shall be the relating_product_definition in exactly one product_definition_usage with a related_product_definition that references a casting_mold_design_element with a role of 'flask'.

WR7: A casting_mold_design_element with the role of 'sand mold component' shall be the relating_product_definition in exactly one product_definition_usage with a related_product_definition that references a casting_mold_design_element with a role of 'pattern and rigging assembly'.

WR8: A casting_mold_design_element with the role of 'pattern and rigging assembly' shall be the relating_product_definition in exactly one assembly_component_usage with a related_product_definition that references a casting_mold_design_element with a role of 'pattern plate'.

WR9: A casting_mold_design_element with the role of 'investment pattern' shall be the relating_product_definition in exactly one product_definition_usage with a related_product_definition that references a die_mold_definition with an application_context_element with a name of 'investment casting'.

5.2.2.1.11 casting_organization_assignment

A casting_organization_assignment relates the organization to a piece of product data.

EXPRESS specification:

```
*)
ENTITY casting_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF organization_assigned_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of document to which an organization is being related.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the casting_organization_assignment entity:

- specification_requires_organization (See ??).

5.2.2.1.12 casting_part_definition

A casting_part_definition is a part that is manufactured via a casting process.

EXPRESS specification:

```
*)
ENTITY casting_part_definition
  SUBTYPE OF (product_definition);
WHERE
  WR1: SIZEOF (QUERY (prpc <* USEDIN (SELF.formation.of_product,
       'METAL_CASTING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
       NOT (prpc.name = 'casting'))) = 0;
  WR2: SELF.frame_of_reference.frame_of_reference.application =
       'casting';
  WR3: SELF.frame_of_reference.name IN ['foundry part',
       'customer part', 'raw', 'finished'];
  WR4: SELF.frame_of_reference.life_cycle_stage IN ['design', 'production'];
  WR5: (NOT (SELF.frame_of_reference.life_cycle_stage = 'production')) OR
       (SELF.frame_of_reference.name IN ['raw', 'finished']);
  WR6: (NOT (SELF.description = 'artifact')) OR
       (SIZEOF (USEDIN (SELF, 'METAL_CASTING.CASTING_DATE_ASSIGNMENT.' +
       'ITEMS')) +
       SIZEOF (USEDIN (SELF, 'METAL_CASTING.' +
       'CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS')) = 1);
END_ENTITY;
(*
```

Formal propositions:

WR1: The casting_part_definition shall be categorized using the string 'casting'.

WR2: The casting_part_definition shall be defined for a casting application.

WR3: The frame of reference for which the casting_part_definition shall be specified is a foundry part, a customer part, a raw part, or a finished part.

WR4: The casting_part_definition represents the part during the design or production portion of its life cycle.

WR5: If the life_cycle_stage of the casting_part_definition is production, the name of the application_context_element shall be raw or finished.

WR6: If the description of the casting_part_defintion is 'artifact', the casting_part_definition shall be referenced by exactly one casting_date_assignment or casting_date_and_time_assignment.

5.2.2.1.13 casting_person_and_organization_assignment

A casting_person_and_organization_assignment relates a person_and_organization to a piece of product data.

EXPRESS specification:

```
*)
ENTITY casting_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF person_and_organization_assigned_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of characterized_object, directed_action, process_plan_version, product_definition, or versioned_action_request to which the person_and_organization is being related.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the casting_person_and_organization_assignment entity:

- design_request_requires_person_and_organization (See **??**);
- design_update_requires_person_and_organization (See **??**).

5.2.2.1.14 casting_round_corner_transition

A casting_round_corner_transition is

EXPRESS specification:

```
*)
ENTITY casting_round_corner_transition
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
       'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') |
       sar.name = 'geometry modification')) = 1;
  WR2: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
```

```
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        pdr.used_representation.name = 'resultant geometry')) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <*
        USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        pdr.used_representation.name = 'feature parameters')) = 1))) = 0;
  WR4: SIZEOF (QUERY (pd <*
        USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
        'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        pdr.used_representation.name = 'feature parameters') |
        NOT ({1 <= SIZEOF (par_rep.used_representation.items) <= 3}))) = 0)))
        = 0;
  WR5: SIZEOF (QUERY (pd <*
        USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
        'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        pdr.used_representation.name = 'feature parameters') |
        NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
        (SIZEOF (['METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
        'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
        'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF (it)) = 3) AND
        (it.name = 'radius') AND
        (SIZEOF (it.qualifiers = 1)) AND
        (SIZEOF (QUERY (qual <* it.qualifiers |
        NOT ('METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF (qual)))) = 0))) <=
        3))) = 0))) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The casting_round_corner_transition shall be the relating_shape_aspect in exactly one shape_aspect_relationship with the name of geometry modification.

WR2: The casting_round_corner_transition shall have exactly one representation with a name of resultant geometry.

WR3: The casting_round_corner_transition shall have exactly one representation with a name of feature parameters.

WR4: The casting_round_corner_transition shall have between one and three representation_items in the items set of the representation with the name of feature parameters.

WR5: The casting_round_corner_transition shall have at most three representation_items with a name of radius in the items set of the representation with the name of feature parameters that are a complex instance of type measure_representation_item, length_measure_with_unit, and qualified_representation_item with exactly one qualifier that is a type_qualifier.

5.2.2.1.15 casting_round_edge_transition

A casting_round_edge_transition is

EXPRESS specification:

```
*)
ENTITY casting_round_edge_transition
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
       'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') |
       sar.name = 'geometry modification')) = 1;
  WR2: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'resultant geometry')) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'feature parameters')) = 1))) = 0;
  WR4: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
       'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'feature parameters') |
       NOT ({1 <= SIZEOF (par_rep.used_representation.items) <= 3}))) = 0)))
       = 0;
  WR5: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
       'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'feature parameters') |
       NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
       (SIZEOF (['METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
       'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
       'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF (it)) = 3) AND
       (it.name = 'radius') AND
       (SIZEOF (it.qualifiers = 1)) AND
       (SIZEOF (QUERY (qual <* it.qualifiers |
       NOT ('METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF (qual)))) = 0))) <=
       3))) = 0))) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The casting_round_edge_transition shall be the relating_shape_aspect in exactly one shape_-

aspect_relationship with the name of geometry modification.

WR2: The casting_round_edge_transition shall have exactly one representation with a name of resultant geometry.

WR3: The casting_round_edge_transition shall have exactly one representation with a name of feature parameters.

WR4: The casting_round_edge_transition shall have between one and three representation_items in the items set of the representation with the name of feature parameters.

WR5: The casting_round_edge_transition shall have at most three representation_items with a name of radius in the items set of the representation with the name of feature parameters that are a complex instance of type measure_representation_item, length_measure_with_unit, and qualified_representation_item with exactly one qualifier that is a type_qualifier.

5.2.2.1.16 casting_taper

A casting_taper is

EXPRESS specification:

```
*)
ENTITY casting_taper
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
       'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') |
       sar.name = 'geometry modification')) = 1;
  WR2: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'resultant geometry')) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'feature parameters')) = 1))) = 0;
  WR4: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
       'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'feature parameters') |
       NOT ({1 <= SIZEOF (par_rep.used_representation.items) <= 7}))) = 0)))
       = 0;
  WR5: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
```

```
      NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
      'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation.name = 'feature parameters') |
      NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
      (SIZEOF (['METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
      'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
      'METAL_CASTING.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF (it)) = 3) AND
      (it.name = 'angle') AND
      (SIZEOF (it.qualifiers = 1)) AND
      (SIZEOF (QUERY (qual <* it.qualifiers |
      NOT ('METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF (qual)))) = 0))) <=
      3))) = 0))) = 0;
  WR6: SIZEOF (QUERY (pd <*
      USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
      'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation.name = 'feature parameters') |
      NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
      (SIZEOF (['METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
      'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
      'METAL_CASTING.RATIO_MEASURE_WITH_UNIT'] * TYPEOF (it)) = 3) AND
      (it.name = 'ratio') AND
      (SIZEOF (it.qualifiers = 1)) AND
      (SIZEOF (QUERY (qual <* it.qualifiers |
      NOT ('METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF (qual)))) = 0))) <=
      3))) = 0))) = 0;
  WR7: SIZEOF (QUERY (pd <*
      USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
      'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation.name = 'feature parameters') |
      NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
      ('METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF (it)) AND
      (it.name = 'taper method') AND
      (it.description IN ['plus', 'minus', 'average']))) = 1))) = 0))) = 0;
  WR8: SIZEOF (QUERY (sar <* USEDIN (SELF,
      'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar.related_shape_aspect.description = 'parting surface')) = 1;
  WR9: SIZEOF (QUERY (sar <* USEDIN (SELF,
      'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar.related_shape_aspect.description = 'reference edge')) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The casting_taper shall be the relating_shape_aspect in exactly one shape_aspect_relationship with the name of geometry modification.

WR2: The casting_taper shall have exactly one representation with a name of resultant geometry.

WR3: The casting_taper shall have exactly one representation with a name of feature parameters.

WR4: The casting_taper shall have between one and seven representation_items in the items set of the representation with the name of feature parameters.

WR5: The casting_taper shall have at most three representation_items with a name of angle in the items set of the representation with the name of feature parameters that are a complex instance of type measure_representation_item, plane_angle_measure_with_unit, and qualified_representation_item with exactly one qualifier that is a type_qualifier.

WR6: The casting_taper shall have at most three representation_items with a name of ratio in the items set of the representation with the name of feature parameters that are a complex instance of type measure_representation_item, ratio_measure_with_unit, and qualified_representation_item with exactly one qualifier that is a type_qualifier.

WR7: The casting_taper shall have exactly one representation_item with a name of taper method in the items set of the representation with the name of feature parameters that is of type descriptive_representation_item with a description value of 'plus', 'minus', or 'average'.

WR8: The casting_taper shall be the relating_shape_aspect in exactly one shape_aspect_relationship in which the related_shape_aspect has a description of parting surface.

WR9: The casting_taper shall be the relating_shape_aspect in exactly one shape_aspect_relationship in which the related_shape_aspect has a description of reference edge.

5.2.2.1.17 change_from_assignment

A change_from_assignment identifies an update that is to be made to a set of one or more characterized_object.

EXPRESS specification:

```
*)
ENTITY change_from_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF change_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of characterized_object to which the action is being related.

5.2.2.1.18 change_to_assignment

A change_to_assignment identifies an update that has been performed on the resulting set of one or more characterized_object.

EXPRESS specification:

```
*)
ENTITY change_to_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF change_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of characterized_object to which the action is being related.

5.2.2.1.19 chemical_structure_element

A chemical_structure_element is .

EXPRESS specification:

```
*)
ENTITY chemical_structure_element
  SUBTYPE OF (product_definition);
WHERE
  WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
       ('METAL_CASTING.PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP' IN
       TYPEOF (pdr)) AND
       (SIZEOF (QUERY (prpc <* USEDIN (pdr.relating_product_definition.
       formation.of_product,
       'METAL_CASTING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
       prpc.name = 'material')) = 1))) >= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: A chemical_structure_element shall be the related_product_definition in at least one product_-definition_relationship that is of type product_material_composition_relationship with a relating_-product_definition that is categorized as a ‘material’.

5.2.2.1.20 circular_runout_tolerance

A circular_runout_tolerance is a type of geometric_tolerance_with_datum_reference which represents the maximum variation of a considered shape_aspect with respect to a fixed position during one complete revolution about a datum_reference.

EXPRESS specification:

```
*)
```

```
ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'circular runout';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system)
       <= 2;
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the circular_runout_tolerance shall be 'circular runout'.

WR2: There shall be at most two datum_references in the datum_system set for the circular_-runout_tolerance.

5.2.2.1.21 concentricity_tolerance

A concentricity_tolerance is a type of geometric_tolerance_with_datum_reference that represents a cylindrical shape_aspect which shall be concentric when rotated about a datum_reference.

EXPRESS specification:

```
*)
ENTITY concentricity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'concentricity';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) =
       1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the concentricity_tolerance shall be 'concentricity'.

WR2: There shall be exactly one datum_reference in the datum_system set for the concentricity_-tolerance.

5.2.2.1.22 data_curve_representation

A data_curve_representation is a representation of a plot of some data of interest that defines the relationship between variables along two axes.

EXPRESS specification:

```
*)
ENTITY data_curve_representation
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF (SELF.items) = 2;
  WR2: SIZEOF (QUERY (it <* SELF.items |
       ('METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF (it)) AND (it.name = 'interpolation method'))) = 1;
  WR3: SIZEOF (QUERY (it <* SELF.items |
       SIZEOF (['METAL_CASTING.CARTESIAN_POINT', 'METAL_CASTING.POLYLINE'] *
       TYPEOF (it)) = 1)) = 1;
  WR4: 'METAL_CASTING.GLOBAL_UNIT_ASSIGNED_CONTEXT' IN TYPEOF
       (SELF.context_of_items);
  WR5: SIZEOF (QUERY (gri <* QUERY (it <* SELF.items |
       'METAL_CASTING.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF (it)) |
       NOT (gri.dim = 2))) = 0;
  WR6: SIZEOF (SELF.context_of_items\global_unit_assigned_context.units) = 1;
  WR7: SIZEOF (QUERY (unit <* SELF.context_of_items\
       global_unit_assigned_context.units |
       NOT (('METAL_CASTING.DERIVED_UNIT' IN TYPEOF (unit)) AND
       (SIZEOF (unit.elements) = 2)))) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: A data_curve_representation shall be comprised of exactly two representation_items in its set of items.

WR2: Exactly one representation_item in the items set shall be a descriptive_representation_item with a name of interpolation method.

WR3: Exactly one representation_item in the items set shall be either a cartesian_point or a polyline.

WR4: A data_curve_representation shall have globally defined units.

WR5: Any geometric_representation_item that comprises a data_curve_representation shall be two dimensional.

WR6: There shall be exactly one globally defined unit for a data_curve_representation.

WR7: Any globally defined unit for a data_curve_representation shall be a derived_unit with exactly two elements.

Informal propositions:

IP1: The first element in the set of elements for a data_curve_representation shall represent the x-axis and the second element shall represent the y-axis. The x exponent shall have the value -1, y exponent shall have the value 1.

5.2.2.1.23 die_mold_definition

A die_mold_definition is the design of the casing into which a substance is poured to make the die cast part.

EXPRESS specification:

```
*)
ENTITY die_mold_definition
  SUBTYPE OF (product_definition);
WHERE
  WR1: SELF.frame_of_reference.name IN ['die casting', 'investment casting'];
  WR2: SELF.frame_of_reference.frame_of_reference.application = 'casting';
  WR3: SELF.frame_of_reference.life_cycle_stage = 'design';
END_ENTITY;
(*
```

Formal propositions:

WR1: The die_mold_definition shall have an application_context_element with a name of 'die casting' or 'investment casting'.

WR2: The die_mold_definition shall have an application_context_element with a frame_of_reference in which the application is 'casting'.

WR3: The die_mold_definition shall have a frame_of_reference in which the life_cycle_stage is 'design'.

5.2.2.1.24 heat_effectivity

A heat_effectivity is .

EXPRESS specification:

```
*)
ENTITY heat_effectivity
  SUBTYPE OF (effectivity);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'METAL_CASTING.CASTING_DATE_ASSIGNMENT.ITEMS')) +
       SIZEOF (USEDIN (SELF, 'METAL_CASTING.' +
       'CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS')) = 1;
  WR2: (NOT (SIZEOF (USEDIN (SELF,
       'METAL_CASTING.CASTING_DATE_ASSIGNMENT.ITEMS')) = 1)) OR
       (SIZEOF (QUERY (cda <* USEDIN (SELF,
       'METAL_CASTING.CASTING_DATE_ASSIGNMENT.ITEMS') |
       NOT (cda.role.name = 'production date'))) = 0);
  WR3: (NOT (SIZEOF (USEDIN (SELF,
       'METAL_CASTING.CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS')) = 1)) OR
       (SIZEOF (QUERY (cdta <* USEDIN (SELF,
```

```
       'METAL_CASTING.CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
       NOT (cdta.role.name = 'production date'))) = 0);
  WR4: (NOT (SIZEOF (USEDIN (SELF,
       'METAL_CASTING.EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY')) >= 1))
       OR (SIZEOF (QUERY (cea <* QUERY (ea <* USEDIN (SELF,
       'METAL_CASTING.EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY') |
       'METAL_CASTING.CASTING_EFFECTIVITY_ASSIGNMENT' IN TYPEOF (ea)) |
       NOT (SIZEOF (QUERY (it <* cea.items |
       NOT (('METAL_CASTING.CASTING_PART_DEFINITION' IN TYPEOF (it)) AND
       (it.description = 'artifact')))) = 0))) = 0);
END_ENTITY;
(*
```

Formal propositions:

WR1: The heat_effectivity shall be used in exactly one casting_date_assignment or casting_date_and_time_assignment.

WR2: If the heat_effectivity is referenced by a date, the role of the reference shall be production date.

WR3: If the heat_effectivity is referenced by a date_and_time, the role of the reference shall be production date.

WR4: If the heat_effectivity is assigned to something, it shall be assigned to a casting_part_definition with a description of artifact.

5.2.2.1.25 identification_marking

An identification_marking is

EXPRESS specification:

```
*)
ENTITY identification_marking
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
       'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') |
       sar.name = 'geometry modification')) = 1;
  WR2: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'resultant geometry')) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <*
       USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd, 'METAL_CASTING.' +
```

```
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation.name = 'feature parameters')) = 1))) = 0;
  WR4: SIZEOF (QUERY (pd <*
      USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
      'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation.name = 'feature parameters') |
      NOT (SIZEOF (par_rep.used_representation.items) = 2))) = 0)))
      = 0;
  WR5: SIZEOF (QUERY (pd <*
      USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
      'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation.name = 'feature parameters') |
      NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
      (SIZEOF (['METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
      'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF (it)) = 2) AND
      (it.name = 'height'))) = 1))) = 0))) = 0;
  WR6: SIZEOF (QUERY (pd <*
      USEDIN (SELF, 'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (par_rep <* QUERY (pdr <* USEDIN (pd,
      'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation.name = 'feature parameters') |
      NOT (SIZEOF (QUERY (it <* par_rep.used_representation.items |
      ('METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF (it)) AND
      (it.name = 'raised') AND
      (it.description IN ['raised', 'not raised']))) = 1))) = 0))) = 0;
  WR7: SIZEOF (QUERY (cdr <*
      USEDIN (SELF, 'METAL_CASTING.CASTING_DOCUMENT_REFERENCE.ITEMS') |
      cdr.assigned_document.kind.product_data_type = 'graphics')) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The identification_marking shall be the relating_shape_aspect in exactly one shape_aspect_relationship with the name of geometry modification.

WR2: The identification_marking shall have exactly one representation with a name of resultant geometry.

WR3: The identification_marking shall have exactly one representation with a name of feature parameters.

WR4: The identification_marking shall have exactly two representation_items in the items set of the representation with the name of feature parameters.

WR5: The identification_marking shall have exactly one representation_item with a name of height in the items set of the representation with the name of feature parameters that is a complex instance of type measure_representation_item and length_measure_with_unit.

WR6: The identification_marking shall have exactly one representation_item with a name of raised in the items set of the representation with the name of feature parameters that is of type descriptive_representation_item with a description of raised or not raised.

WR7: The identification_marking shall be in the items set of exactly one casting_document_reference that assigns a document with a product_data_type of graphics.

5.2.2.1.26 investment_mold_definition

An investment_mold_definition is the design of the casing into which a substance is poured to make the investment cast part.

EXPRESS specification:

```
*)
ENTITY investment_mold_definition
  SUBTYPE OF (product_definition);
WHERE
  WR1: SELF.frame_of_reference.name = 'investment casting';
  WR2: SELF.frame_of_reference.frame_of_reference.application = 'casting';
  WR3: SELF.frame_of_reference.life_cycle_stage = 'design';
END_ENTITY;
(*
```

Formal propositions:

WR1: The investment_mold_definition shall have an application_context_element with a name of 'investment casting'.

WR2: The investment_mold_definition shall have an application_context_element with a frame_of_reference in which the application is 'casting'.

WR3: The investment_mold_definition shall have a frame_of_reference in which the life_cycle_stage is 'design'.

5.2.2.1.27 investment_mold_pattern_definition

An investment_mold_pattern_definition is the design of

EXPRESS specification:

```
*)
ENTITY investment_mold_pattern_definition
  SUBTYPE OF (product_definition);
WHERE
  WR1: SELF.frame_of_reference.name = 'investment casting';
  WR2: SELF.frame_of_reference.frame_of_reference.application =
       'casting';
```

```
  WR3: SELF.frame_of_reference.life_cycle_stage = 'design';
  WR4: SIZEOF (QUERY (pdu <* QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
       'METAL_CASTING.PRODUCT_DEFINITION_USAGE' IN
       TYPEOF (pdr)) |
       NOT ('METAL_CASTING.INVESTMENT_MOLD_DEFINITION' IN
       TYPEOF (pdu.relating_product_definition)))) = 0;
  WR5: SIZEOF (QUERY (acu <* QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
       'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN
       TYPEOF (pdr)) |
       'METAL_CASTING.INVESTMENT_MOLD_SPRUE_DEFINITION' IN
       TYPEOF (acu.related_product_definition))) = 1;
  WR6: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
       ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
       TYPEOF (pdr.related_product_definition)) AND
       (pdr.related_product_definition.description = 'investment pattern')))
       >= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The investment_mold_pattern_definition shall be specified for use in an investment casting.

WR2: The investment_mold_pattern_definition shall be specified for use in a casting application.

WR3: The investment_mold_pattern_definition shall be defined in the design life cycle.

WR4: The investment_mold_pattern_definition shall be the related_product_definition in product_definition_relationships of type product_definition_usage only with a relating_product_definition that is of type investment_mold_definition.

WR5: The investment_mold_pattern_definition shall be the relating_product_definition in exactly one product_definition_relationship of type assembly_component_usage with a relating_product_definition that is of type investment_mold_sprue_definition.

WR6: The investment_mold_pattern_definition shall be the relating_product_definition in at least one product_definition_relationship with a related_product_definition that is of type casting_mold_design_element with a description of investment pattern.

5.2.2.1.28 investment_mold_sprue_definition

An investment_mold_sprue_definition is the design of .

EXPRESS specification:

```
*)
```

```
ENTITY investment_mold_sprue_definition
  SUBTYPE OF (product_definition);
WHERE
  WR1: SELF.frame_of_reference.name = 'investment casting';
  WR2: SELF.frame_of_reference.frame_of_reference.application =
       'casting';
  WR3: SELF.frame_of_reference.life_cycle_stage = 'design';
  WR4: SIZEOF (QUERY (acu <* QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
       'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN
       TYPEOF (pdr)) |
       ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
       TYPEOF (acu.related_product_definition)) AND
       (acu.description = 'sprue component'))) >= 1;
  WR5: SIZEOF (QUERY (acu <* QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
       'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN
       TYPEOF (pdr)) |
       'METAL_CASTING.INVESTMENT_MOLD_PATTERN_DEFINITION' IN
       TYPEOF (acu.relating_product_definition))) >= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The investment_mold_sprue_definition shall be specified for use in an investment casting.

WR2: The investment_mold_sprue_definition shall be specified for use in a casting application.

WR3: The investment_mold_sprue_definition shall be defined in the design life cycle.

WR4: The investment_mold_sprue_definition shall be the relating_product_definition in at least one product_definition_relationship of type assembly_component_usage with a related_product_definition that is of type casting_mold_design_element with a description of sprue component.

WR5: The investment_mold_sprue_definition shall be the related_product_definition in at least one product_definition_relationship of type assembly_component_usage with a relating_product_definition that is of type investment_mold_pattern_definition.

5.2.2.1.29 linear_profile_tolerance

A linear_profile_tolerance is a type of geometric_tolerance that defines the allowable deviation of a curve that may require datum_reference to control the definition.

EXPRESS specification:

```
*)
ENTITY linear_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
```

```
WHERE
  WR1: SELF\geometric_tolerance.name = 'linear profile';
  WR2: (NOT ('FEATURE_BASED_PROCESS_PLANNING.' +
       'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF))) OR
       (SIZEOF
       (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the linear_profile_tolerance shall be 'linear profile'.

WR2: There shall be at three or less datum_references in the datum_system set for the linear_profile_tolerance.

5.2.2.1.30 parallelism_tolerance

A parallelism_tolerance is a type of geometric_tolerance_with_datum_reference that defines the allowable deviation of a shape_aspect that may require datum_reference to control the definition.

EXPRESS specification:

```
*)
ENTITY parallelism_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'parallelism';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) =
       1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the parallelism_tolerance shall be 'parallelism'.

WR2: There shall be exactly one datum_reference in the datum_system set for the parallelism_tolerance.

5.2.2.1.31 perpendicularity_tolerance

A perpendicularity_tolerance is a type of geometric_tolerance_with_datum_reference that defines the allowable deviation of a shape_aspect that may require datum_reference to control the definition.

EXPRESS specification:

```
*)
ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'perpendicularity';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system)
       = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the perpendicularity_tolerance shall be 'perpendicularity'.

WR2: There shall be exactly one datum_reference in the datum_system set for the perpendicularity_tolerance.

5.2.2.1.32 physical_structure_element

A physical_structure_element is .

EXPRESS specification:

```
*)
ENTITY physical_structure_element
  SUBTYPE OF (product_definition);
WHERE
  WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
       ('METAL_CASTING.PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP' IN
       TYPEOF (pdr)) AND
       (SIZEOF (QUERY (prpc <* USEDIN (pdr.relating_product_definition.
       formation.of_product,
       'METAL_CASTING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
       prpc.name = 'material')) = 1))) >= 1;
  WR2: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
       NOT (pdr.relating_product_definition.formation :=:
       pdr.related_product_definition.formation))) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: A physical_structure_element shall be the related_product_definition in at least one product_-definition_relationship that is of type product_material_composition_relationship with a relating_-product_definition that is categorized as a 'material'.

WR2: Every product_definition_relationship in which the physical_structure_element is the relating_product_definition shall have a related_product_definition that references the same instance of product_definition_formation.

5.2.2.1.33 position_tolerance

A position_tolerance is a type of geometric_tolerance_with_datum_reference that defines the allowable deviation of a shape_aspect that may require datum_reference to control the definition.

EXPRESS specification:

```
*)
ENTITY position_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'position';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <=
       3;
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the position_tolerance shall be 'position'.

WR2: There shall be at most three datum_references in the datum_system set for the position_tolerance.

5.2.2.1.34 process_plan_input_assignment

A process_plan_input_assignment assigns a set of items to a particular process_plan_version or step in a process_plan_version as input to that step or version.

EXPRESS specification:

```
*)
ENTITY process_plan_input_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF process_plan_input_item;

WHERE
  WR1: 'METAL_CASTING.PROCESS_PLAN_VERSION' IN TYPEOF (SELF.assigned_action);
END_ENTITY;
(*
```

Attribute definitions:

items: the set of characterized_object or product_definition that are defined as input to the process_plan_version or a step of the process_plan_version.

Formal propositions:

WR1: The assigned_action of the process_plan_input_assignment shall be of type process_plan_version.

5.2.2.1.35 process_plan_version

A process_plan_version is a variant of a procedure used to manufacture a part using a casting process.

EXPRESS specification:

```
*)
ENTITY process_plan_version
  SUBTYPE OF (action);
WHERE

  WR1: SIZEOF (USEDIN (SELF, 'METAL_CASTING.CASTING_DATE_ASSIGNMENT.ITEMS')) +
       SIZEOF (USEDIN (SELF, 'METAL_CASTING.' +
       'CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS')) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The process_plan_version shall be referenced by exactly one casting_date_assignment or casting_date_and_time_assignment.

5.2.2.1.36 property_inspection_or_test_requirement

A property_inspection_or_test_requirement is a property_definition which defines the property for which a test or inspection is necessary.

EXPRESS specification:

```
*)
ENTITY property_inspection_or_test_requirement
  SUBTYPE OF (property_definition);
WHERE
  WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
       pdr.name = 'inspection or test basis')) = 1;
  WR2: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_RELATIONSHIP.' +
       'RELATED_PROPERTY_DEFINITION') |
       NOT (pdr.relating_property_definition.description IN
       ['composition requirement', 'shape requirement',
       'tolerance requirement', 'process requirement',
```

```
       'heat treat process requirement', 'property requirement',
       'required machining allowance', 'surface roughness requirement'])))
       = 0;
  WR3: SIZEOF (USEDIN (SELF, 'CASTING_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1;
  WR4: (NOT (SELF.name = 'sampling run size')) OR
       (('METAL_CASTING.CASTING_PART_DEFINITION' IN TYPEOF (SELF.definition))
       AND (SELF.definition.frame_of_reference.name IN
       ['customer part', 'foundary part']));
END_ENTITY;
(*
```

Formal propositions:

WR1: The property_inspection_or_test_requirement shall be used in exactly one property_definition_relationship with a name of 'inspection or test basis' as the related_property_definition.

WR2: When the property_inspection_or_test_requirement is used as the related_property_definition in a property_definition_relationship, the relating_property_definition shall have a description of 'composition requirement', 'shape requirement', 'tolerance requirement', 'process requirement', 'heat treat process requirement', or 'property requirement'.

WR3: A property_inspection_or_test_requirement shall have a representation.

WR4: If the name of the property_inspection_or_test_requirement is 'sampling run size', the requirement shall be for a customer part or a foundary part.

5.2.2.1.37 referenced_property_exception

A referenced_property_exception is .

EXPRESS specification:

```
*)
ENTITY referenced_property_exception
  SUBTYPE OF (property_definition);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'METAL_CASTING.' +
       'CASTING_DOCUMENT_REFERENCE.ITEMS')) >= 1;
  WR2: SIZEOF (QUERY (pdr <* USEDIN (SELF,
       'METAL_CASTING.PROPERTY_DEFINITION_RELATIONSHIP.' +
       'RELATED_PROPERTY_DEFINITION') |
       NOT ((pdr.relating_property_definition.description IN
       ['composition requirement', 'shape requirement',
       'tolerance requirement', 'process requirement',
       'heat treat process requirement', 'property requirement',
       'required machining allowance', 'surface roughness requirement']) OR
       (SIZEOF (TYPEOF (pdr.relating_property_definition) *
       ['METAL_CASTING.PROPERTY_INSPECTION_OR_TEST_REQUIREMENT',
```

```
      'METAL_CASTING.REPORTING_REQUIREMENT']) = 1)))) = 0;
  WR3: SIZEOF (USEDIN (SELF, 'CASTING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1;
  WR4: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'CASTING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      NOT (SIZEOF (pdr.used_representation.items) = 1))) = 0;
  WR5: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'CASTING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      NOT (SIZEOF (QUERY (it <* pdr.used_representation.items |
      NOT ('METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF (it)))) = 0))) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: A referenced_property_exception shall be referenced by at least one document.

WR2: When the referenced_property_exception is used as the related_property_definition in a property_definition_relationship, the relating_property_definition shall have a description of 'composition requirement', 'shape requirement', 'tolerance requirement', 'process requirement', 'heat treat process requirement', 'property requirement' 'required machining allowance', or 'surface roughness requirement' or shall be a property_inspection_or_test_requirement or reporting_requirement.

WR3: A referenced_property_exception shall have exactly one representation.

WR4: A referenced_property_exception shall have a representation with exactly one representation_item.

WR5: The representation_item for a referenced_property_exception shall be a descriptive_representation_item.

5.2.2.1.38 reporting_requirement

A reporting_requirement is ..

EXPRESS specification:

```
*)
ENTITY reporting_requirement
  SUBTYPE OF (property_definition);
WHERE
  WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
      pdr.name = 'reporting basis')) = 1;
  WR2: SIZEOF (USEDIN (SELF, 'CASTING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1;
  WR3: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.' +
```

```
      'RELATED_PROPERTY_DEFINITION') |
      NOT ((pdr.relating_property_definition.description IN
      ['composition requirement', 'shape requirement',
      'tolerance requirement', 'process requirement',
      'heat treat process requirement', 'property requirement',
      'required machining allowance', 'surface roughness requirement']) OR
      ('METAL_CASTING.PROPERTY_INSPECTION_OR_TEST_REQUIREMENT' IN
      TYPEOF (pdr.relating_property_definition))))) = 0;
  WR4: ('METAL_CASTING.CASTING_PART_DEFINITION' IN TYPEOF (SELF.definition))
      AND (SELF.definition.frame_of_reference.name IN
      ['foundary part', 'customer part', 'finished']);

END_ENTITY;
(*
```

Formal propositions:

WR1: The reporting_requirement shall be used in exactly one property_definition_relationship with a name of 'reporting basis' as the related_property_definition.

WR2: A reporting_requirement shall have a representation.

WR3: When the reporting_requirement is used as the related_property_definition in a property_-definition_relationship, the relating_property_definition shall have a description of 'composition requirement', 'shape requirement', 'tolerance requirement', 'process requirement', 'heat treat process requirement', or 'property requirement' or shall be a property_inspection_or_test_requirement.

WR4: The reporting_requirement shall be defined for a casting_part_definition that is , a customer part, foundary part or finished part.

5.2.2.1.39 sand_mold_definition

A sand_mold_definition is the design of the casing into which a substance is poured to make the sand cast part. A sand_mold_definition may be comprised of other sand_mold_definitions.

EXPRESS specification:

```
*)
ENTITY sand_mold_definition
  SUBTYPE OF (product_definition);
WHERE
  WR1: SELF.frame_of_reference.name = 'sand casting';
  WR2: SELF.frame_of_reference.frame_of_reference.application =
      'casting';
  WR3: SELF.frame_of_reference.life_cycle_stage = 'design';
  WR4: {1 <= SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
      ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
      TYPEOF (pdr.related_product_definition)) AND
```

```
      (pdr.related_product_definition.description =
      'sand mold component'))) <= 2};
  WR5: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
      (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
      TYPEOF (pdr.related_product_definition)) AND
      (pdr.related_product_definition.description =
      'sand mold component')) AND
      (NOT (pdr.name IN ['cope', 'drag'])))) = 0;
  WR6: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
      ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
      TYPEOF (pdr.related_product_definition)) AND
      (pdr.related_product_definition.description = 'flask'))) = 1;
  WR7: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
      ('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN
      TYPEOF (pdr.related_product_definition)) AND
      (pdr.related_product_definition.description =
      'pattern and rigging assembly'))) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The sand_mold_definition shall be specified for use in a sand casting.

WR2: The sand_mold_definition shall be specified for use in a casting application.

WR3: The sand_mold_definition shall be defined in the design life cycle.

WR4: The sand_mold_definition shall be the relating_product_definition in one or two product_definition_relationships with a related_product_definition that is a casting_mold_design_element with a description of 'sand mold component'.

WR5: If the sand_mold_definition is the relating_product_definition in a product_definition_relationship with a related_product_definition that is a sand mold component, the name of the product_definition_relationship shall be either 'cope' or 'drag'.

WR6: The sand_mold_definition shall be the relating_product_definition in exactly one product_definition_relationship with a related_product_definition that is a casting_mold_design_element with a description of 'flask'.

WR7: The sand_mold_definition shall be the relating_product_definition in exactly one product_definition_relationship with a related_product_definition that is a casting_mold_design_element with a description of 'pattern and rigging assembly'.

5.2.2.1.40 simulation_input

A simulation_input is .

EXPRESS specification:

```
*)
ENTITY simulation_input
  SUBTYPE OF (property_definition);
WHERE
  WR1: SIZEOF (QUERY (caa <* USEDIN (SELF,
       'METAL_CASTING.CASTING_ACTION_ASSIGNMENT.ITEMS') |
       'METAL_CASTING.SIMULATION_RUN' IN TYPEOF (caa.assigned_action))) >= 1;
  WR2: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
       'METAL_CASTING.SIMULATION_INPUT_REGION' IN
       TYPEOF (pdr.related_property_definition))) >= 1;
  WR4: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       pdr.used_representation.name = 'integration interval')) >= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The simulation_input shall be used in at least one simulation_run.

WR2: The simulation_input shall be the relating_property_definition in at least one property_definition_relationship with a related_property_definition of type simulation_input_region.

WR3: The simulation_input shall have at least one representation with a name of integration interval.

Informal propositions:

IP1: The ordering of the integration intervals shall be satisfied by the use of the representation_relationship entity. The preceding integration interval shall be specified in the representation that is referenced by the rep1 attribute of the representation_relationship. The succeeding integration interval shall be specified in the representation that is referenced by the rep2 attribute of the representation_relationship.

5.2.2.1.41 simulation_input_region

A simulation_input_region is .

EXPRESS specification:

```
*)
ENTITY simulation_input_region
  SUBTYPE OF (property_definition);
WHERE
  WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
        ('METAL_CASTING.SHAPE_REPRESENTATION' IN
        TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'region shape'))) = 1;
  WR2: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        SIZEOF (QUERY (it <* pdr.used_representation.items |
        ('METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN  TYPEOF (it)) AND
        (it.name = 'region type'))) = 1)) = 1;
  WR3: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
        'METAL_CASTING.SIMULATION_INPUT' IN
        TYPEOF (pdr.relating_property_definition))) >= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The simulation_input_region shall have exactly one representation that is a shape_representation with a name of region shape.

WR2: The simulation_input_region shall have exactly one representation with exactly one representation_item with a name of region type in its items set that is of type descriptive_representation_item.

WR3: The simulation_input_region shall be the related_property_definition in at least one property_definition_relationship with a relating_property_definition of type simulation_input.

5.2.2.1.42 simulation_output_region

A simulation_output_region is .

EXPRESS specification:

```
*)
ENTITY simulation_output_region
  SUBTYPE OF (property_definition);
WHERE
  WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
        'METAL_CASTING.SIMULATION_INPUT_REGION' IN
        TYPEOF (pdr.relating_property_definition))) = 1;
  WR2: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('METAL_CASTING.SHAPE_REPRESENTATION' IN
        TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'region shape'))) = 1;
  WR3: SIZEOF (QUERY (pdr <* USEDIN (SELF, 'METAL_CASTING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'METAL_CASTING.SIMULATION_UNIT_STATE' IN
        TYPEOF (pdr.used_representation))) >= 1;
```

```
END_ENTITY;
(*
```

Formal propositions:

WR1: The simulation_output_region shall be the related_property_definition in at least one property_definition_relationship with a relating_property_definition of type simulation_input_region.

WR2: The simulation_output_region shall have exactly one representation that is a shape_representation with a name of region shape.

WR3: The simulation_output_region shall have exactly one representation that is of type simulation_unit_state.

5.2.2.1.43 simulation_run

A simulation_run is a single execution of a computer application that simulates the process of producing the cast part.

EXPRESS specification:

```
*)
ENTITY simulation_run
  SUBTYPE OF (action);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'METAL_CASTING.ACTION_RESOURCE.USAGE')) = 1;
  WR2: SIZEOF (QUERY (ar <* USEDIN (SELF,
       'METAL_CASTING.ACTION_RESOURCE.USAGE') |
       NOT (ar.kind.name = 'simulation software'))) = 0;

  WR3: SIZEOF (QUERY (caa <* QUERY (aa <* USEDIN (SELF,
       'METAL_CASTING.ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
       'METAL_CASTING.CASTING_ACTION_ASSIGNMENT' IN TYPEOF (aa)) |
       SIZEOF (QUERY (it <* caa.items |
       'METAL_CASTING.SIMULATION_INPUT' IN TYPEOF (it))) = 1)) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The simulation_run shall be used by exactly one action_resource.

WR2: The simulation_run shall be used by an action_resource with a type name of simulation software.

WR3: The simulation_run shall have exactly one input as specified by a simulation_input.

5.2.2.1.44 simulation_unit_state

A simulation_unit_state is .

EXPRESS specification:

```
*)
ENTITY simulation_unit_state
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF (QUERY (it <* SELF.items |
       (SIZEOF (['METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
       'METAL_CASTING.TIME_MEASURE_WITH_UNIT'] * TYPEOF (it)) = 2) AND
       (it.name = 'evaluation time'))) = 1;
  WR2: SIZEOF (QUERY (it <* SELF.items |
       'METAL_CASTING.TENSOR_REPRESENTATION_ITEM' IN TYPEOF (it))) >= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The simulation_unit_state shall have exactly one representation_item with a name of evaluation time in its items set that is a complex instance of type measure_representation_item and time_measure_with_unit.

WR2: The simulation_unit_state shall have at least one representation_item in its items set that is of type tensor_representation_item.

5.2.2.1.45 statistical_measure

A statistical_measure is .

EXPRESS specification:

```
*)
ENTITY statistical_measure
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF (SELF.items) = 3;
  WR2: SIZEOF (QUERY (it <* SELF.items |
       ('METAL_CASTING.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it)) AND
       (it.name = 'mean'))) = 1;
  WR3: SIZEOF (QUERY (it <* SELF.items |
       ('METAL_CASTING.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it)) AND
       (it.name = 'variance'))) = 1;
  WR4: SIZEOF (QUERY (it <* SELF.items |
       ('METAL_CASTING.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it)) AND
       ('METAL_CASTING.COUNT_MEASURE' IN
       TYPEOF (it\measure_with_unit.value_component)) AND
```

```
       (it.name = 'number of measurements'))) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The statistical_measure shall have exactly three representation_items in its items set.

WR2: The statistical_measure shall have exactly one representation_item with a name of mean in its items set that is of type measure_representation_item.

WR3: The statistical_measure shall have exactly one representation_item with a name of variance in its items set that is of type measure_representation_item.

WR4: The statistical_measure shall have exactly one representation_item with a name of number of measurements in its items set that is of type measure_representation_item with a value_component of type count_measure.

5.2.2.1.46 surface_profile_tolerance

A surface_profile_tolerance is a type of geometric_tolerance that defines the allowable deviation of a shape_aspect that may require datum_reference to control the definition.

EXPRESS specification:

```
*)
ENTITY surface_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: SELF\geometric_tolerance.name = 'surface profile';
  WR2: (NOT ('FEATURE_BASED_PROCESS_PLANNING.' +
       'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE')) OR (SIZEOF
       (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the surface_profile_tolerance shall be 'surface profile'.

WR2: There shall be three or less datum_references in the datum_system set for the surface_profile_-tolerance.

5.2.2.1.47 symmetry_tolerance

A symmetry_tolerance is a type of geometric_tolerance_with_datum_reference that defines the allowable deviation of a shape_aspect that may require datum_reference to control the definition.

EXPRESS specification:

```
*)
ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'symmetry';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <=
       3;
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the symmetry_tolerance shall be 'symmetry'.

WR2: There shall be at most three datum_references in the datum_system set for the symmetry_tolerance.

5.2.2.1.48 tensor_representation_item

A tensor_representation_item is a representation element that may be either a scalar measure, a vector measure, or a tensor measure.

EXPRESS specification:

```
*)
ENTITY tensor_representation_item
  SUBTYPE OF (representation_item);
  tensor_value : tensor_type;
END_ENTITY;
(*
```

Attribute definitions:

tensor_value: the scalar, tensor1, tensor2_2d, or tensor2_3d that gives the value for the tensor_representation_item.

5.2.2.1.49 total_runout_tolerance

A total_runout_tolerance is a type of geometric_tolerance_with_datum_reference that defines the allowable deviation of a shape_aspect that may require datum_reference to control the definition.

EXPRESS specification:

```
*)
ENTITY total_runout_tolerance
```

```
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'total runout';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <=
       2;
END_ENTITY;
(*
```

Formal propositions:

WR1: The name for the total_runout_tolerance shall be 'total runout'.

WR2: There shall be at most two datum_references in the datum_system set for the total_runout_tolerance.

5.2.2.2 Exchange of design and manufacturing product information for cast parts imported entity modifications

5.2.2.2.1 action_method

The base definition of the action_method entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the action_method entity:

- process_execution_record_requires_process (See **??**).

5.2.2.2.2 action_relationship

The base definition of the action_relationship entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the action_relationship entity:

- process_execution_requires_process (See **??**).

5.2.2.2.3 application_context

The base definition of the application_context entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the application_context entity:

- application_context_requires_ap_definition (See **??**).

5.2.2.2.4 application_protocol_definition

The base definition of the application_protocol_definition entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the application_protocol_definition entity:

- application_context_requires_ap_definition (See **??**).

5.2.2.2.5 approval

The base definition of the approval entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the approval entity:

- approval_requires_approval_date_time (See **??**);
- approval_requires_approval_person_organization (See **??**);
- dependent_instantiable_approval (See **??**).

5.2.2.2.6 approval_date_time

The base definition of the approval_date_time entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the approval_date_time entity:

– approval_requires_approval_date_time (See ??).

5.2.2.2.7 approval_person_organization

The base definition of the approval_person_organization entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the approval_person_organization entity:

– approval_requires_approval_person_organization (See ??).

5.2.2.2.8 directed_action

The base definition of the directed_action entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the directed_action entity:

– design_update_requires_date_or_date_and_time (See ??);

– design_update_requires_person_and_organization (See ??).

5.2.2.2.9 document

The base definition of the document entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the document entity:

– specification_requires_organization (See ??).

5.2.2.2.10 geometric_tolerance

The base definition of the geometric_tolerance entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the geometric_tolerance entity:

– geometric_tolerance_subtype_exclusiveness (See **??**).

5.2.2.2.11 lot_effectivity

The base definition of the lot_effectivity entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the lot_effectivity entity:

– lot_requires_date_or_date_and_time (See **??**).

5.2.2.2.12 product_definition_process

The base definition of the product_definition_process entity is given in ISO 10303-49. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the product_definition_process entity:

– process_execution_requires_process (See **??**).

5.2.2.2.13 property_process

The base definition of the property_process entity is given in ISO 10303-49. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the property_process entity:

– process_execution_requires_process (See ??).

5.2.2.2.14 property_definition

The base definition of the property_definition entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the property_definition entity:

– item_requirement_requires_specification (See ??).

5.2.2.2.15 representation

The base definition of the representation entity is given in ISO 10303-43. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the representation entity:

– annotation_requires_shape_aspect (See ??);

– process_execution_record_requires_process (See ??).

5.2.2.2.16 shape_aspect

The base definition of the shape_aspect entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the shape_aspect entity:

– annotation_requires_shape_aspect (See ??).

5.2.2.2.17 versioned_action_request

The base definition of the versioned_action_request entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the versioned_action_request entity:

- design_request_requires_date_or_date_and_time (See **??**);
- design_request_requires_person_and_organization (See **??**).

5.2.3 Exchange of design and manufacturing product information for cast parts rule definitions

5.2.3.1 annotation_requires_shape_aspect

The annotation_requires_shape_aspect rule specifies that every representation with the name of 'shape aspect annotation' shall be assigned to exactly one shape_aspect.

EXPRESS specification:

```
*)
RULE annotation_requires_shape_aspect FOR
     (representation, shape_aspect);
WHERE
  WR1: SIZEOF (QUERY (rep <* representation |
       (NOT (rep.name = 'shape aspect annotation')) OR
       (SIZEOF (QUERY (sa <* shape_aspect |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (rep,
       'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.' +
       'USED_REPRESENTATION') |
       sa :=: pdr.definition.definition)) = 1))) = 0))) = 0;
END_RULE;
(*
```

Attribute definitions:

representation: the set of all instances of representation.

shape_aspect: the set of all instances of shape_aspect.

Formal propositions:

WR1: Each representation that is an annotation for a shape_aspect shall be used as the used_representation in exactly one property_definition_representation for any shape_aspect.

5.2.3.2 application_context_requires_ap_definition

The application_context_requires_ap_definition rule specifies that each instance of application_context shall be referenced by exactly one application_protocol_definition that specifies this part of ISO 10303.

EXPRESS specification:

```
*)
RULE application_context_requires_ap_definition FOR
  (application_context, application_protocol_definition);
WHERE
  WR1: SIZEOF (QUERY (ac <* application_context |
       NOT (SIZEOF (QUERY (apd <* application_protocol_definition |
       (ac :=: apd.application)
       AND
       (apd.application_interpreted_model_schema_name =
       'metal_casting'))) = 1 ))) = 0;
END_RULE;
(*
```

Attribute definitions:

application_context: the set of all instances of application_context entities.

application_protocol_definition: the set of all instances of application_protocol_definition entities.

Formal propositions:

WR1: For each instance of application_context, there shall be exactly one instance of application_protocol_definition that references the instance of application_context as its application with a value of 'metal_casting' as its application_interpreted_model_schema_name.

5.2.3.3 approval_requires_approval_date_time

The approval_requires_approval_date_time rule specifies that each instance of approval shall be referenced by exactly one approval_date_time. This rule enforces the requirement for every approval to have a date or a date and time on which the approval obtained its specified status.

EXPRESS specification:

```
*)
RULE approval_requires_approval_date_time FOR (approval,
  approval_date_time);
WHERE
  WR1: SIZEOF (QUERY (app <* approval |
       NOT (SIZEOF (QUERY (adt <* approval_date_time |
```

```
       app :=: adt.dated_approval )) = 1 ))) = 0;
END_RULE;
(*
```

Attribute definitions:

approval: the set of all instances of approval entities.

approval_date_time: the set of all instances of approval_date_time entities.

Formal propositions:

WR1: For each instance of approval, there shall be exactly one instance of approval_date_time which contains the instance of approval as its dated_approval attribute.

5.2.3.4 approval_requires_approval_person_organization

The approval_requires_approval_person_organization specifies that each instance of approval shall have exactly one approval_person_organization referencing it. This rule enforces the requirement for an approval to be authorized by a person within an organization.

EXPRESS specification:

```
*)
RULE approval_requires_approval_person_organization FOR
  (approval, approval_person_organization);
WHERE
  WR1: SIZEOF (QUERY (app <* approval |
       NOT (SIZEOF (QUERY (apo <* approval_person_organization |
       app :=: apo.authorized_approval )) = 1 ))) = 0;
END_RULE;
(*
```

Attribute definitions:

approval: the set of all instances of approval entities.

approval_person_organization: the set of all instances of approval_person_organization entities.

Formal propositions:

WR1: For each instance of approval, there shall be one or more instances of approval_person_organization which contains the instance of approval as its authorized_approval attribute.

5.2.3.5 dependent_instantiable_approval

The dependent_instantiable_approval rule specifies that all instances of approval are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_approval FOR (approval);
WHERE
  WR1: SIZEOF (QUERY (ast <* approval |
       NOT (SIZEOF (USEDIN (ast, '')) >= 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

approval: the set of all instances of approval.

Formal propositions:

WR1: For each instance of approval, there shall be a reference to the approval instance from an attribute of another entity.

5.2.3.6 design_request_requires_date_or_date_and_time

The design_request_requires_date_or_date_and_time rule specifies that every versioned_action_request shall be referenced by exactly one date or date_and_time.

EXPRESS specification:

```
*)
RULE design_request_requires_date_or_date_and_time FOR
     (casting_date_assignment, casting_date_and_time_assignment,
      versioned_action_request);
WHERE
  WR1: SIZEOF (QUERY (vr <* versioned_action_request |
       NOT (SIZEOF (QUERY (cda <*
       casting_date_assignment |
       vr IN cda.items)) +
       SIZEOF (QUERY (cdta <*
       casting_date_assignment |
       vr IN cdta.items)) = 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

casting_date_and_time_assignment: the set of all instances of casting_date_and_time_assignment.

casting_date_assignment: the set of all instances of casting_date_assignment.

versioned_action_request: the set of all instances of versioned_action_request.

Formal propositions:

WR1: Each versioned_action_request shall be in the set of items of exactly one casting_date_-assignment or casting_- date_and_time_assignment.

5.2.3.7 design_request_requires_person_and_organization

The design_request_requires_person_and_organization rule specifies that every versioned_action_request shall be referenced by exactly one person_and_organization.

EXPRESS specification:

```
*)
RULE design_request_requires_person_and_organization FOR
     (casting_person_and_organization_assignment, versioned_action_request);
WHERE
  WR1: SIZEOF (QUERY (vr <* versioned_action_request |
       NOT (SIZEOF (QUERY (cpoa <*
       casting_person_and_organization_assignment |
       vr IN cpoa.items)) = 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

casting_person_and_organization_assignment: the set of all instances of casting_person_and_-organization_- assignment.

versioned_action_request: the set of all instances of versioned_action_request.

Formal propositions:

WR1: Each versioned_action_request shall be in the set of items of exactly one casting_person_-and_organization_- assignment.

5.2.3.8 design_update_requires_date_or_date_and_time

The design_update_requires_date_or_date_and_time rule specifies that every directed_action that specifies a 'design update' shall be referenced by exactly one date or date_and_time.

EXPRESS specification:

```
*)
```

```
RULE design_update_requires_date_or_date_and_time FOR
     (casting_date_assignment, casting_date_and_time_assignment,
      directed_action);
WHERE
  WR1: SIZEOF (QUERY (da <* directed_action |
       NOT (SIZEOF (QUERY (cda <*
       casting_date_assignment |
       (NOT (da.name = 'design update')) OR
       (da IN cda.items))) +
       SIZEOF (QUERY (cdta <*
       casting_date_assignment |
       (NOT (da.name = 'design update')) OR
       (da IN cdta.items))) = 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

casting_date_and_time_assignment: the set of all instances of casting_date_and_time_assignment.

casting_date_assignment: the set of all instances of casting_date_assignment.

directed_action: the set of all instances of directed_action.

Formal propositions:

WR1: Each directed_action that is a design update shall be in the set of items of exactly one casting_date_assignment or casting_date_and_time_assignment.

5.2.3.9 design_update_requires_item_definition

The design_update_requires_item_definition rule specifies that every casting_action_assignment that assigns a 'design update' shall be assigned to exactly one item definition.

EXPRESS specification:

```
*)
RULE design_update_requires_item_definition FOR
     (casting_action_assignment);
WHERE
  WR1: SIZEOF (QUERY (caa <* casting_action_assignment |
       (caa.assigned_action.name = 'design update') AND
       (SIZEOF (caa.items) = 1) AND
       (SIZEOF (['METAL_CASTING.CHARACTERIZED_OBJECT',
       'METAL_CASTING.PRODUCT_DEFINITION',
       'METAL_CASTING.PROPERTY_DEFINITION',
       'METAL_CASTING.SHAPE_ASPECT'] * TYPEOF (caa.items)) = 1))) = 1;
END_RULE;
(*
```

Attribute definitions:

casting_action_assignment: the set of all instances of casting_action_assignment.

Formal propositions:

WR1: Each casting_action_assignment that assigns a design update shall reference exactly one characterized_object, product_definition, property_definition, or shape_aspect in the set of items.

5.2.3.10 design_update_requires_person_and_organization

The design_update_requires_person_and_organization rule specifies that every directed_action that specifies a 'design update' shall be referenced by exactly one person_and_organization.

EXPRESS specification:

```
*)
RULE design_update_requires_person_and_organization FOR
     (casting_person_and_organization_assignment, directed_action);
WHERE
  WR1: SIZEOF (QUERY (da <* directed_action |
       NOT (SIZEOF (QUERY (cpoa <*
       casting_person_and_organization_assignment |
       (NOT (da.name = 'design update')) OR
       (da IN cpoa.items))) = 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

casting_person_and_organization_assignment: the set of all instances of casting_person_and_-organization_- assignment.

directed_action: the set of all instances of directed_action.

Formal propositions:

WR1: Each directed_action that is a design update shall be in the set of items of exactly one casting_person_and_- organization_assignment.

5.2.3.11 geometric_tolerance_subtype_exclusiveness

The geometric_tolerance_subtype_exclusiveness rule specifies that an instance of the subtypes of a geometric_- tolerance shall be only one of angularity_tolerance, circular_runout_tolerance, concentricity_tolerance, linear_- profile_tolerance, parallelism_tolerance, perpendicularity_tolerance, positional_tolerance, surface_profile_- tolerance, symmetry_tolerance, or total_runout_tolerance.

EXPRESS specification:

```
*)
RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);
WHERE
  WR1: SIZEOF (QUERY (gt <* geometric_tolerance |
       SIZEOF (TYPEOF (gt) *
       ['METAL_CASTING.ANGULARITY_TOLERANCE',
       'METAL_CASTING.CIRCULAR_RUNOUT_TOLERANCE',
       'METAL_CASTING.CONCENTRICITY_TOLERANCE',
       'METAL_CASTING.LINEAR_PROFILE_TOLERANCE',
       'METAL_CASTING.PARALLELISM_TOLERANCE',
       'METAL_CASTING.PERPENDICULARITY_TOLERANCE',
       'METAL_CASTING.POSITIONAL_TOLERANCE',
       'METAL_CASTING.SURFACE_PROFILE_TOLERANCE',
       'METAL_CASTING.SYMMETRY_TOLERANCE',
       'METAL_CASTING.TOTAL_RUNOUT_TOLERANCE'])
       >= 2)) = 0;
END_RULE;
(*
```

Attribute definitions:

geometric_tolerance: the set of all instances of geometric_tolerance.

Formal propositions:

WR1: Each instance of the subtypes of geometric_tolerance shall be one of circular_runout_tolerance, angularity_- tolerance, concentricity_tolerance, parallelism_tolerance, perpendicularity_tolerance, total_runout_tolerance, surface_profile_tolerance, symmetry_tolerance, positional_tolerance, or linear_-profile_tolerance.

5.2.3.12 item_requirement_requires_specification

The item_requirement_requires_specification rule specifies that every property_definition shall be given in at least one document.

EXPRESS specification:

```
*)
RULE item_requirement_requires_specification FOR
     (property_definition, casting_document_reference);
WHERE
  WR1: SIZEOF (QUERY (pd <* property_definition |
       NOT (SIZEOF (QUERY (cdr <* casting_document_reference |
       (NOT ((pd.description IN
       ['composition requirement', 'shape requirement',
       'tolerance requirement', 'process requirement',
       'heat treat process requirement', 'property requirement',
```

```
        'required machining allowance', 'surface roughness requirement']) OR
        (SIZEOF (TYPEOF (pd) *
        ['METAL_CASTING.PROPERTY_INSPECTION_OR_TEST_REQUIREMENT',
        'METAL_CASTING.REPORTING_REQUIREMENT']) = 1) OR
        (pd IN cdr.items))))) >= 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

casting_document_reference: the set of all instances of casting_document_reference.

property_definition: the set of all instances of property_definition.

Formal propositions:

WR1: Each property_definition that defines an item requirement shall be in the set of items of at least one casting_- document_reference.

5.2.3.13 lot_requires_date_or_date_and_time

The lot_requires_date_or_date_and_time rule specifies that every lot_effectivity shall be referenced by exactly one date or date_and_time.

EXPRESS specification:

```
*)
RULE lot_requires_date_or_date_and_time FOR
     (casting_date_assignment, casting_date_and_time_assignment,
      lot_effectivity);
WHERE
  WR1: SIZEOF (QUERY (le <* lot_effectivity |
       NOT (SIZEOF (QUERY (cda <*
       casting_date_assignment |
       le IN cda.items)) +
       SIZEOF (QUERY (cdta <*
       casting_date_assignment |
       le IN cdta.items)) = 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

casting_date_and_time_assignment: the set of all instances of casting_date_and_time_assignment.

casting_date_assignment: the set of all instances of casting_date_assignment.

lot_effectivity: the set of all instances of lot_effectivity.

Formal propositions:

WR1: Each lot_effectivity shall be in the set of items of exactly one casting_date_assignment or casting_date_and_- time_assignment.

5.2.3.14 process_execution_record_requires_process

The process_execution_record_requires_process rule specifies that every representation with a name of 'machine setting record', 'substance usage record', 'substance composition element record', or 'process parameter record' shall be used to define exactly one property of a process.

EXPRESS specification:

```
*)
RULE process_execution_record_requires_process FOR
     (representation, action_method);
WHERE
  WR1: SIZEOF (QUERY (rep <* representation |
       (NOT (rep.name IN ['machine setting record', 'substance usage record',
       'substance composition element record',
       'process parameter record'])) OR
       (SIZEOF (QUERY (am <* action_method |
       NOT (SIZEOF (QUERY (apr <* USEDIN (rep,
       'METAL_CASTING.ACTION_PROPERTY_REPRESENTATION.REPRESENTATION') |
       am :=: apr.property)) = 1))) = 0))) = 0;
END_RULE;
(*
```

Attribute definitions:

action_method: the set of all instances of action_method.

representation: the set of all instances of representation.

Formal propositions:

WR1: Each representation that is a process execution record shall be used as the representation in exactly one action_property_representation for a process.

5.2.3.15 process_execution_requires_process_plan

The process_execution_requires_process_plan rule specifies that every process_execution_effectivity shall be referenced by exactly one date or date_and_time.

EXPRESS specification:

```
*)
```

```
RULE process_execution_requires_process_plan FOR
     (product_definition_process, property_process,
      action_relationship);
WHERE
  WR1: SIZEOF (QUERY (pdp <* product_definition_process |
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       ('METAL_CASTING.EXECUTED_ACTION' IN TYPEOF (pdp)) AND
       (pdp :=: ar.related_action))) = 1))) = 0;
  WR2: SIZEOF (QUERY (ppp <* property_process |
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       ('METAL_CASTING.EXECUTED_ACTION' IN TYPEOF (ppp)) AND
       (ppp :=: ar.related_action))) = 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

action_relationship: the set of all instances of action_relationship.

product_definition_process: the set of all instances of product_definition_process.

property_process: the set of all instances of product_property_process.

Formal propositions:

WR1: Each product_definition_process that is also of type executed_action shall be the related_-product_definition in exactly on action_relationship.

WR2: Each property_process that is also of type executed_action shall be the related_product_-definition in exactly on action_relationship.

5.2.3.16 specification_requires_organization

The specification_requires_organization rule specifies that every document of type 'specification' shall be assigned to exactly one organization.

EXPRESS specification:

```
*)
RULE specification_requires_organization FOR
     (casting_organization_assignment, document);
WHERE
  WR1: SIZEOF (QUERY (doc <* document |
       NOT (SIZEOF (QUERY (coa <* casting_organization_assignment |
       (NOT (doc.kind.product_data_type = 'specification')) OR
       (doc IN coa.items))) = 1))) = 0;
END_RULE;
(*
```

Attribute definitions:

casting_organization_assignment: the set of all instances of casting_organization_assignment.

document: the set of all instances of document.

Formal propositions:

WR1: Each document that is a specification shall be in the set of items of exactly one casting_-organization_- assignment.

6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation method(s) supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods:

- ISO 10303-21.

Requirements with respect to implementation methods are specified in Annex C.

The Protocol Information Conformance Statement (PICS) proforma lists the options or the combinations of options that may be included in the implementation. The PICS proforma is provided in annex D.

This Part of ISO 10303 provides for a number of options that may be supported by an implementation. These options have been grouped into the following conformance classes:

Support for a particular conformance class requires support of all the options specified in this class.

NOTE – ISO 10303-323 defines the abstract test suite to be used in the assessment of conformance. ISO 10303-32 describes the conformance assessment process.

Annex A
(normative)

AIM EXPRESS expanded listing

EXPRESS specification:

```
*)
SCHEMA metal_casting;

  TYPE action_assigned_item = SELECT
    (casting_part_definition,
     characterized_object,
     product_definition,
     property_definition,
     shape_aspect,
     simulation_input);
  END_TYPE; -- action_assigned_item

  TYPE action_request_assigned_item = SELECT
    (product_definition,
     product_definition_formation,
     property_definition,
     shape_aspect);
  END_TYPE; -- action_request_assigned_item

  TYPE ahead_or_behind = ENUMERATION OF
    (ahead,
     behind);
  END_TYPE; -- ahead_or_behind

  TYPE anisotropic_symmetric_tensor2_2d = ARRAY [1:3] OF
 context_dependent_measure;
  END_TYPE; -- anisotropic_symmetric_tensor2_2d

  TYPE anisotropic_symmetric_tensor2_3d = ARRAY [1:6] OF
 context_dependent_measure;
  END_TYPE; -- anisotropic_symmetric_tensor2_3d

  TYPE approval_assigned_item = SELECT
    (product_definition_formation,
     process_plan_version);
  END_TYPE; -- approval_assigned_item

  TYPE axis2_placement = SELECT
    (axis2_placement_2d,
     axis2_placement_3d);
  END_TYPE; -- axis2_placement

  TYPE b_spline_curve_form = ENUMERATION OF
    (polyline_form,
```

```
   circular_arc,
   elliptic_arc,
   parabolic_arc,
   hyperbolic_arc,
   unspecified);
END_TYPE; -- b_spline_curve_form

TYPE b_spline_surface_form = ENUMERATION OF
  (plane_surf,
   cylindrical_surf,
   conical_surf,
   spherical_surf,
   toroidal_surf,
   surf_of_revolution,
   ruled_surf,
   generalised_cone,
   quadric_surf,
   surf_of_linear_extrusion,
   unspecified);
END_TYPE; -- b_spline_surface_form

TYPE boolean_operand = SELECT
  (solid_model);
END_TYPE; -- boolean_operand

TYPE change_item = SELECT
  (characterized_object);
END_TYPE; -- change_item

TYPE characterized_action_definition = SELECT
  (action,
   action_method,
   action_method_relationship,
   action_relationship);
END_TYPE; -- characterized_action_definition

TYPE characterized_definition = SELECT
  (characterized_object,
   characterized_product_definition,
   shape_definition);
END_TYPE; -- characterized_definition

TYPE characterized_material_property = SELECT
  (product_material_composition_relationship);
END_TYPE; -- characterized_material_property

TYPE characterized_product_definition = SELECT
  (product_definition,
   product_definition_relationship);
END_TYPE; -- characterized_product_definition

TYPE characterized_resource_definition = SELECT
```

```
  (action_resource,
   action_resource_requirement);
END_TYPE; -- characterized_resource_definition

TYPE context_dependent_measure = REAL;
END_TYPE; -- context_dependent_measure

TYPE count_measure = NUMBER;
END_TYPE; -- count_measure

TYPE curve_on_surface = SELECT
  (pcurve,
   surface_curve,
   composite_curve_on_surface);
END_TYPE; -- curve_on_surface

TYPE date_and_time_assigned_item = SELECT
  (casting_part_definition,
   directed_action,
   heat_effectivity,
   lot_effectivity,
   process_plan_version,
   representation,
   versioned_action_request);
END_TYPE; -- date_and_time_assigned_item

TYPE date_assigned_item = SELECT
  (action_resource,
   casting_part_definition,
   directed_action,
   heat_effectivity,
   lot_effectivity,
   process_plan_version,
   representation,
   versioned_action_request);
END_TYPE; -- date_assigned_item

TYPE date_time_select = SELECT
  (date,
   local_time,
   date_and_time);
END_TYPE; -- date_time_select

TYPE day_in_month_number = INTEGER;
END_TYPE; -- day_in_month_number

TYPE dimension_count = INTEGER;
WHERE
  wr1: SELF > 0;
END_TYPE; -- dimension_count

TYPE dimensional_characteristic = SELECT
```

```
  (dimensional_location,
   dimensional_size);
END_TYPE; -- dimensional_characteristic

TYPE document_assigned_item = SELECT
  (action_method,
   action_resource,
   action_resource_requirement,
   dimensional_location,
   dimensional_size,
   geometric_tolerance,
   identification_marking,
   material_property,
   product_definition_process,
   property_process,
   property_definition,
   representation,
   simulation_input);
END_TYPE; -- document_assigned_item

TYPE effectivity_assigned_item = SELECT
  (casting_part_definition);
END_TYPE; -- effectivity_assigned_item

TYPE geometric_set_select = SELECT
  (point,
   curve,
   surface);
END_TYPE; -- geometric_set_select

TYPE hour_in_day = INTEGER;
WHERE
  wr1: (0 <= SELF) AND (SELF < 24);
END_TYPE; -- hour_in_day

TYPE identifier = STRING;
END_TYPE; -- identifier

TYPE isotropic_symmetric_tensor2_3d = context_dependent_measure;
END_TYPE; -- isotropic_symmetric_tensor2_3d

TYPE knot_type = ENUMERATION OF
  (uniform_knots,
   unspecified,
   quasi_uniform_knots,
   piecewise_bezier_knots);
END_TYPE; -- knot_type

TYPE label = STRING;
END_TYPE; -- label

TYPE length_measure = REAL;
```

```
END_TYPE; -- length_measure

TYPE list_of_reversible_topology_item = LIST [0:?] OF
            reversible_topology_item;
END_TYPE; -- list_of_reversible_topology_item

TYPE measure_value = SELECT
  (length_measure,
   time_measure,
   plane_angle_measure,
   parameter_value,
   context_dependent_measure,
   positive_length_measure,
   count_measure);
END_TYPE; -- measure_value

TYPE minute_in_hour = INTEGER;
WHERE
  wr1: (0 <= SELF) AND (SELF <= 59);
END_TYPE; -- minute_in_hour

TYPE month_in_year_number = INTEGER;
WHERE
  wr1: (1 <= SELF) AND (SELF <= 12);
END_TYPE; -- month_in_year_number

TYPE organization_assigned_item = SELECT
  (document);
END_TYPE; -- organization_assigned_item

TYPE orthotropic_symmetric_tensor2_3d = ARRAY [1:3] OF
            context_dependent_measure;
END_TYPE; -- orthotropic_symmetric_tensor2_3d

TYPE parameter_value = REAL;
END_TYPE; -- parameter_value

TYPE pcurve_or_surface = SELECT
  (pcurve,
   surface);
END_TYPE; -- pcurve_or_surface

TYPE person_and_organization_assigned_item = SELECT
  (characterized_object,
   directed_action,
   process_plan_version,
   product_definition,
   versioned_action_request);
END_TYPE; -- person_and_organization_assigned_item

TYPE person_organization_select = SELECT
  (person,
```

```
   organization,
   person_and_organization);
END_TYPE; -- person_organization_select

TYPE plane_angle_measure = REAL;
END_TYPE; -- plane_angle_measure

TYPE positive_length_measure = length_measure;
WHERE
  wr1: SELF > 0;
END_TYPE; -- positive_length_measure

TYPE preferred_surface_curve_representation = ENUMERATION OF
  (curve_3d,
   pcurve_s1,
   pcurve_s2);
END_TYPE; -- preferred_surface_curve_representation

TYPE process_or_process_relationship = SELECT
  (product_definition_process,
   property_process,
   relationship_with_condition);
END_TYPE; -- process_or_process_relationship

TYPE process_plan_input_item = SELECT
  (characterized_object,
   product_definition);
END_TYPE; -- process_plan_input_item

TYPE property_or_shape_select = SELECT
  (property_definition,
   shape_definition);
END_TYPE; -- property_or_shape_select

TYPE relationship_with_condition = SELECT
  (action_method_relationship,
   action_relationship);
END_TYPE; -- relationship_with_condition

TYPE reversible_topology = SELECT
  (reversible_topology_item,
   list_of_reversible_topology_item,
   set_of_reversible_topology_item);
END_TYPE; -- reversible_topology

TYPE reversible_topology_item = SELECT
  (edge,
   path,
   face,
   face_bound,
   closed_shell,
   open_shell);
```

```
END_TYPE; -- reversible_topology_item

TYPE scalar = context_dependent_measure;
END_TYPE; -- scalar

TYPE second_in_minute = REAL;
WHERE
  wr1: (0 <= SELF) AND (SELF < 60);
END_TYPE; -- second_in_minute

TYPE set_of_reversible_topology_item = SET [0:?] OF
            reversible_topology_item;
END_TYPE; -- set_of_reversible_topology_item

TYPE shape_definition = SELECT
  (product_definition_shape,
   shape_aspect,
   shape_aspect_relationship);
END_TYPE; -- shape_definition

TYPE shell = SELECT
  (open_shell,
   closed_shell);
END_TYPE; -- shell

TYPE supported_item = SELECT
  (action_directive,
   action,
   action_method);
END_TYPE; -- supported_item

TYPE symmetric_tensor2_2d = SELECT
  (anisotropic_symmetric_tensor2_2d);
END_TYPE; -- symmetric_tensor2_2d

TYPE symmetric_tensor2_3d = SELECT
  (isotropic_symmetric_tensor2_3d,
   orthotropic_symmetric_tensor2_3d,
   anisotropic_symmetric_tensor2_3d);
END_TYPE; -- symmetric_tensor2_3d

TYPE tensor1 = SELECT
  (tensor1_2d,
   tensor1_3d);
END_TYPE; -- tensor1

TYPE tensor1_2d = ARRAY [1:2] OF context_dependent_measure;
END_TYPE; -- tensor1_2d

TYPE tensor1_3d = ARRAY [1:3] OF context_dependent_measure;
END_TYPE; -- tensor1_3d
```

```
TYPE tensor_type = SELECT
  (scalar,
   tensor1,
   symmetric_tensor2_2d,
   symmetric_tensor2_3d,
   anisotropic_symmetric_tensor2_2d,
   anisotropic_symmetric_tensor2_3d,
   isotropic_symmetric_tensor2_3d,
   orthotropic_symmetric_tensor2_3d);
END_TYPE; -- tensor_type

TYPE text = STRING;
END_TYPE; -- text

TYPE time_measure = REAL;
END_TYPE; -- time_measure

TYPE tolerance_select = SELECT
  (geometric_tolerance);
END_TYPE; -- tolerance_select

TYPE transformation = SELECT
  (functionally_defined_transformation);
END_TYPE; -- transformation

TYPE transition_code = ENUMERATION OF
  (discontinuous,
   continuous,
   cont_same_gradient,
   cont_same_gradient_same_curvature);
END_TYPE; -- transition_code

TYPE trimming_preference = ENUMERATION OF
  (cartesian,
   parameter,
   unspecified);
END_TYPE; -- trimming_preference

TYPE trimming_select = SELECT
  (cartesian_point,
   parameter_value);
END_TYPE; -- trimming_select

TYPE unit = SELECT
  (named_unit,
   derived_unit);
END_TYPE; -- unit

TYPE value_qualifier = SELECT
  (type_qualifier);
END_TYPE; -- value_qualifier
```

```
TYPE vector_or_direction = SELECT
  (vector,
   direction);
END_TYPE; -- vector_or_direction

TYPE year_number = INTEGER;
END_TYPE; -- year_number

ENTITY action;
    name          : label;
    description   : text;
    chosen_method : action_method;
END_ENTITY; -- action

ENTITY action_assignment
  ABSTRACT SUPERTYPE;
    assigned_action : action;
END_ENTITY; -- action_assignment

ENTITY action_directive;
    name        : label;
    description : text;
    analysis    : text;
    comment     : text;
    requests    : SET [1:?] OF versioned_action_request;
END_ENTITY; -- action_directive

ENTITY action_method;
    name        : label;
    description : text;
    consequence : text;
    purpose     : text;
END_ENTITY; -- action_method

ENTITY action_method_relationship;
    name            : label;
    description     : text;
    relating_method : action_method;
    related_method  : action_method;
END_ENTITY; -- action_method_relationship

ENTITY action_property;
    name        : label;
    description : text;
    definition  : characterized_action_definition;
END_ENTITY; -- action_property

ENTITY action_property_representation;
    name           : label;
    description    : text;
    property       : action_property;
    representation : representation;
```

```
END_ENTITY; -- action_property_representation

ENTITY action_relationship;
    name            : label;
    description     : text;
    relating_action : action;
    related_action  : action;
END_ENTITY; -- action_relationship

ENTITY action_request_assignment
  ABSTRACT SUPERTYPE;
    assigned_action_request : versioned_action_request;
END_ENTITY; -- action_request_assignment

ENTITY action_request_solution;
    method  : action_method;
    request : versioned_action_request;
END_ENTITY; -- action_request_solution

ENTITY action_resource;
    name        : label;
    description : text;
    usage       : SET [1:?] OF supported_item;
    kind        : action_resource_type;
END_ENTITY; -- action_resource

ENTITY action_resource_requirement;
    name        : label;
    description : text;
    kind        : resource_requirement_type;
    operations  : SET [1:?] OF characterized_action_definition;
END_ENTITY; -- action_resource_requirement

ENTITY action_resource_type;
    name : label;
END_ENTITY; -- action_resource_type

ENTITY advanced_brep_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
             'METAL_CASTING.MANIFOLD_SOLID_BREP',
             'METAL_CASTING.FACETED_BREP','METAL_CASTING.MAPPED_ITEM',
             'METAL_CASTING.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1)) )) =
              0;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
             'METAL_CASTING.MANIFOLD_SOLID_BREP',
             'METAL_CASTING.MAPPED_ITEM'] * TYPEOF(it)) = 1) )) > 0;
    wr3: SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
             'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (NOT
             (SIZEOF(QUERY ( csh <* msb_shells(msb,'AIC_ADVANCED_BREP')
              | (NOT (SIZEOF(QUERY ( fcs <* csh\connected_face_set.
```

```
                cfs_faces | (NOT ('METAL_CASTING.ADVANCED_FACE' IN TYPEOF(
                fcs))) )) = 0)) )) = 0)) )) = 0;
      wr4: SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
                'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
                'METAL_CASTING.ORIENTED_CLOSED_SHELL' IN TYPEOF(msb\
                manifold_solid_brep.outer)) )) = 0;
      wr5: SIZEOF(QUERY ( brv <* QUERY ( it <* SELF.items | (
                'METAL_CASTING.BREP_WITH_VOIDS' IN TYPEOF(it)) ) | (NOT (
                SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids | csh\
                oriented_closed_shell.orientation )) = 0)) )) = 0;
      wr6: SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
                'METAL_CASTING.MAPPED_ITEM' IN TYPEOF(it)) ) | (NOT (
                'METAL_CASTING.ADVANCED_BREP_SHAPE_REPRESENTATION' IN
                TYPEOF(mi\mapped_item.mapping_source.mapped_representation))) ))
                = 0;
  END_ENTITY; -- advanced_brep_shape_representation

  ENTITY advanced_face
    SUBTYPE OF (face_surface);
    WHERE
      wr1 : SIZEOF(['METAL_CASTING.ELEMENTARY_SURFACE',
                 'METAL_CASTING.B_SPLINE_SURFACE',
                 'METAL_CASTING.SWEPT_SURFACE'] * TYPEOF(face_geometry)) = 1;
      wr2 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
                 'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (NOT (
                 SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list | (NOT
                 ('METAL_CASTING.EDGE_CURVE' IN TYPEOF(oe.edge_element))) ))
                 = 0)) )) = 0;
      wr3 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
                 'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (NOT (
                 SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list | (NOT
                 (SIZEOF(['METAL_CASTING.LINE','METAL_CASTING.CONIC',
                 'METAL_CASTING.POLYLINE','METAL_CASTING.SURFACE_CURVE',
                 'METAL_CASTING.B_SPLINE_CURVE'] * TYPEOF(oe.edge_element\
                 edge_curve.edge_geometry)) = 1)) )) = 0)) )) = 0;
      wr4 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
                 'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (NOT (
                 SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list | (NOT
                 (('METAL_CASTING.VERTEX_POINT' IN TYPEOF(oe.edge_start))
                 AND ('METAL_CASTING.CARTESIAN_POINT' IN TYPEOF(oe.
                 edge_start\vertex_point.vertex_geometry)) AND (
                 'METAL_CASTING.VERTEX_POINT' IN TYPEOF(oe.edge_end)) AND (
                 'METAL_CASTING.CARTESIAN_POINT' IN TYPEOF(oe.edge_end\
                 vertex_point.vertex_geometry)))) )) = 0)) )) = 0;
      wr5 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
                 'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (
                 'METAL_CASTING.ORIENTED_PATH' IN TYPEOF(elp_fbnds.bound)) ))
                 = 0;
      wr6 : (NOT ('METAL_CASTING.SWEPT_SURFACE' IN TYPEOF(face_geometry)))
                 OR (SIZEOF(['METAL_CASTING.LINE','METAL_CASTING.CONIC',
                 'METAL_CASTING.POLYLINE','METAL_CASTING.B_SPLINE_CURVE'] *
                 TYPEOF(face_geometry\swept_surface.swept_curve)) = 1);
```

```
    wr7 : SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* SELF.bounds | (
             'METAL_CASTING.VERTEX_LOOP' IN TYPEOF(bnds.bound)) ) | (
             NOT (('METAL_CASTING.VERTEX_POINT' IN TYPEOF(vlp_fbnds\
             face_bound.bound\vertex_loop.loop_vertex)) AND (
             'METAL_CASTING.CARTESIAN_POINT' IN TYPEOF(vlp_fbnds\
             face_bound.bound\vertex_loop.loop_vertex\vertex_point.
             vertex_geometry)))) )) = 0;
    wr8 : SIZEOF(QUERY ( bnd <* SELF.bounds | (NOT (SIZEOF([
             'METAL_CASTING.EDGE_LOOP','METAL_CASTING.VERTEX_LOOP'] *
             TYPEOF(bnd.bound)) = 1)) )) = 0;
    wr9 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
             'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (NOT (
             SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list | ((
             'METAL_CASTING.SURFACE_CURVE' IN TYPEOF(oe.edge_element\
             edge_curve.edge_geometry)) AND (NOT (SIZEOF(
             QUERY ( sc_ag <* oe.edge_element\edge_curve.edge_geometry\
             surface_curve.associated_geometry | (NOT (
             'METAL_CASTING.PCURVE' IN TYPEOF(sc_ag))) )) = 0))) )) = 0)) ))
             = 0;
    wr10: ((NOT ('METAL_CASTING.SWEPT_SURFACE' IN TYPEOF(face_geometry)))
             OR (NOT ('METAL_CASTING.POLYLINE' IN TYPEOF(face_geometry\
             swept_surface.swept_curve))) OR (SIZEOF(face_geometry\
             swept_surface.swept_curve\polyline.points) < 3)) AND (
             SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
             'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (NOT (
             SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list | ((
             'METAL_CASTING.POLYLINE' IN TYPEOF(oe.edge_element\
             edge_curve.edge_geometry)) AND (NOT (SIZEOF(oe.edge_element
             \edge_curve.edge_geometry\polyline.points) < 3))) )) = 0)) ))
             = 0);
END_ENTITY; -- advanced_face

ENTITY angularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SELF\geometric_tolerance.name = 'angularity';
    wr2: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
            datum_system) = 1;
END_ENTITY; -- angularity_tolerance

ENTITY application_context;
    application : text;
  INVERSE
    context_elements : SET [1:?] OF application_context_element FOR
                          frame_of_reference;
END_ENTITY; -- application_context

ENTITY application_context_element
  SUPERTYPE OF (ONEOF (product_context,product_definition_context));
    name               : label;
    frame_of_reference : application_context;
END_ENTITY; -- application_context_element
```

```
ENTITY application_protocol_definition;
    status                                   : label;
    application_interpreted_model_schema_name : label;
    application_protocol_year                : year_number;
    application                              : application_context;
END_ENTITY; -- application_protocol_definition

ENTITY approval;
    status : approval_status;
    level  : label;
END_ENTITY; -- approval

ENTITY approval_assignment
  ABSTRACT SUPERTYPE;
    assigned_approval : approval;
END_ENTITY; -- approval_assignment

ENTITY approval_date_time;
    date_time      : date_time_select;
    dated_approval : approval;
END_ENTITY; -- approval_date_time

ENTITY approval_person_organization;
    person_organization : person_organization_select;
    authorized_approval : approval;
    role                : approval_role;
END_ENTITY; -- approval_person_organization

ENTITY approval_role;
    role : label;
END_ENTITY; -- approval_role

ENTITY approval_status;
    name : label;
END_ENTITY; -- approval_status

ENTITY assembly_component_usage
  SUBTYPE OF (product_definition_usage);
    reference_designator : OPTIONAL identifier;
END_ENTITY; -- assembly_component_usage

ENTITY axis1_placement
  SUBTYPE OF (placement);
    axis : OPTIONAL direction;
  DERIVE
    z : direction := NVL(normalise(axis),direction([0,0,1]));
  WHERE
    wr1: SELF\geometric_representation_item.dim = 3;
END_ENTITY; -- axis1_placement

ENTITY axis2_placement_2d
```

```
  SUBTYPE OF (placement);
    ref_direction : OPTIONAL direction;
  DERIVE
    p : LIST [2:2] OF direction := build_2axes(ref_direction);
  WHERE
    wr1: SELF\geometric_representation_item.dim = 2;
END_ENTITY; -- axis2_placement_2d

ENTITY axis2_placement_3d
  SUBTYPE OF (placement);
    axis          : OPTIONAL direction;
    ref_direction : OPTIONAL direction;
  DERIVE
    p : LIST [3:3] OF direction := build_axes(axis,ref_direction);
  WHERE
    wr1: SELF\placement.location.dim = 3;
    wr2: (NOT EXISTS(axis)) OR (axis.dim = 3);
    wr3: (NOT EXISTS(ref_direction)) OR (ref_direction.dim = 3);
    wr4: (NOT EXISTS(axis)) OR (NOT EXISTS(ref_direction)) OR (
             cross_product(axis,ref_direction).magnitude > 0);
END_ENTITY; -- axis2_placement_3d

ENTITY b_spline_curve
  SUPERTYPE OF (ONEOF (uniform_curve,b_spline_curve_with_knots,
      quasi_uniform_curve,bezier_curve) ANDOR rational_b_spline_curve)
  SUBTYPE OF (bounded_curve);
    degree              : INTEGER;
    control_points_list : LIST [2:?] OF cartesian_point;
    curve_form          : b_spline_curve_form;
    closed_curve        : LOGICAL;
    self_intersect      : LOGICAL;
  DERIVE
    upper_index_on_control_points : INTEGER := SIZEOF(
                                       control_points_list) - 1;
    control_points                : ARRAY [0:
                                       upper_index_on_control_points] OF
                                        cartesian_point := list_to_array(
                                       control_points_list,0,
                                       upper_index_on_control_points);
  WHERE
    wr1: ('METAL_CASTING.UNIFORM_CURVE' IN TYPEOF(SELF)) OR (
             'METAL_CASTING.QUASI_UNIFORM_CURVE' IN TYPEOF(SELF)) OR (
             'METAL_CASTING.BEZIER_CURVE' IN TYPEOF(SELF)) OR (
             'METAL_CASTING.B_SPLINE_CURVE_WITH_KNOTS' IN TYPEOF(SELF));
END_ENTITY; -- b_spline_curve

ENTITY b_spline_curve_with_knots
  SUBTYPE OF (b_spline_curve);
    knot_multiplicities : LIST [2:?] OF INTEGER;
    knots               : LIST [2:?] OF parameter_value;
    knot_spec           : knot_type;
  DERIVE
```

```
    upper_index_on_knots : INTEGER := SIZEOF(knots);
  WHERE
    wr1: constraints_param_b_spline(degree,upper_index_on_knots,
             upper_index_on_control_points,knot_multiplicities,knots);
    wr2: SIZEOF(knot_multiplicities) = upper_index_on_knots;
END_ENTITY; -- b_spline_curve_with_knots

ENTITY b_spline_surface
  SUPERTYPE OF (ONEOF (b_spline_surface_with_knots,uniform_surface,
      quasi_uniform_surface,bezier_surface) ANDOR
      rational_b_spline_surface)
  SUBTYPE OF (bounded_surface);
    u_degree            : INTEGER;
    v_degree            : INTEGER;
    control_points_list : LIST [2:?] OF LIST [2:?] OF cartesian_point;
    surface_form        : b_spline_surface_form;
    u_closed            : LOGICAL;
    v_closed            : LOGICAL;
    self_intersect      : LOGICAL;
  DERIVE
    u_upper        : INTEGER := SIZEOF(control_points_list) - 1;
    v_upper        : INTEGER := SIZEOF(control_points_list[1]) - 1;
    control_points : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF
                       cartesian_point := make_array_of_array(
                      control_points_list,0,u_upper,0,v_upper);
  WHERE
    wr1: ('METAL_CASTING.UNIFORM_SURFACE' IN TYPEOF(SELF)) OR (
             'METAL_CASTING.QUASI_UNIFORM_SURFACE' IN TYPEOF(SELF)) OR (
             'METAL_CASTING.BEZIER_SURFACE' IN TYPEOF(SELF)) OR (
             'METAL_CASTING.B_SPLINE_SURFACE_WITH_KNOTS' IN TYPEOF(SELF));
END_ENTITY; -- b_spline_surface

ENTITY b_spline_surface_with_knots
  SUBTYPE OF (b_spline_surface);
    u_multiplicities : LIST [2:?] OF INTEGER;
    v_multiplicities : LIST [2:?] OF INTEGER;
    u_knots          : LIST [2:?] OF parameter_value;
    v_knots          : LIST [2:?] OF parameter_value;
    knot_spec        : knot_type;
  DERIVE
    knot_u_upper : INTEGER := SIZEOF(u_knots);
    knot_v_upper : INTEGER := SIZEOF(v_knots);
  WHERE
    wr1: constraints_param_b_spline(SELF\b_spline_surface.u_degree,
             knot_u_upper,SELF\b_spline_surface.u_upper,u_multiplicities,
             u_knots);
    wr2: constraints_param_b_spline(SELF\b_spline_surface.v_degree,
             knot_v_upper,SELF\b_spline_surface.v_upper,v_multiplicities,
             v_knots);
    wr3: SIZEOF(u_multiplicities) = knot_u_upper;
    wr4: SIZEOF(v_multiplicities) = knot_v_upper;
END_ENTITY; -- b_spline_surface_with_knots
```

```
ENTITY bezier_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- bezier_curve

ENTITY bezier_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- bezier_surface

ENTITY bounded_curve
  SUPERTYPE OF (ONEOF (polyline,b_spline_curve,trimmed_curve,
      composite_curve))
  SUBTYPE OF (curve);
END_ENTITY; -- bounded_curve

ENTITY bounded_surface
  SUPERTYPE OF (b_spline_surface)
  SUBTYPE OF (surface);
END_ENTITY; -- bounded_surface

ENTITY brep_with_voids
  SUBTYPE OF (manifold_solid_brep);
    voids : SET [1:?] OF oriented_closed_shell;
END_ENTITY; -- brep_with_voids

ENTITY calendar_date
  SUBTYPE OF (date);
    day_component   : day_in_month_number;
    month_component : month_in_year_number;
  WHERE
    wr1: valid_calendar_date(SELF);
END_ENTITY; -- calendar_date

ENTITY cartesian_point
  SUBTYPE OF (point);
    coordinates : LIST [1:3] OF length_measure;
END_ENTITY; -- cartesian_point

ENTITY cartesian_transformation_operator
  SUPERTYPE OF (cartesian_transformation_operator_3d)
  SUBTYPE OF (geometric_representation_item,
      functionally_defined_transformation);
    axis1        : OPTIONAL direction;
    axis2        : OPTIONAL direction;
    local_origin : cartesian_point;
    scale        : OPTIONAL REAL;
  DERIVE
    scl : REAL := NVL(scale,1);
  WHERE
    wr1: scl > 0;
END_ENTITY; -- cartesian_transformation_operator
```

```
ENTITY cartesian_transformation_operator_3d
  SUBTYPE OF (cartesian_transformation_operator);
    axis3 : OPTIONAL direction;
  DERIVE
    u : LIST [3:3] OF direction := base_axis(3,SELF\
           cartesian_transformation_operator.axis1,SELF\
           cartesian_transformation_operator.axis2,axis3);
  WHERE
    wr1: SELF\cartesian_transformation_operator.dim = 3;
END_ENTITY; -- cartesian_transformation_operator_3d

ENTITY casting_action_assignment
  SUBTYPE OF (action_assignment);
    items : SET [1:?] OF action_assigned_item;
END_ENTITY; -- casting_action_assignment

ENTITY casting_action_request_assignment
  SUBTYPE OF (action_request_assignment);
    items : SET [1:?] OF action_request_assigned_item;
END_ENTITY; -- casting_action_request_assignment

ENTITY casting_approval_assignment
  SUBTYPE OF (approval_assignment);
    items : SET [1:?] OF approval_assigned_item;
END_ENTITY; -- casting_approval_assignment

ENTITY casting_date_and_time_assignment
  SUBTYPE OF (date_and_time_assignment);
    items : SET [1:?] OF date_and_time_assigned_item;
END_ENTITY; -- casting_date_and_time_assignment

ENTITY casting_date_assignment
  SUBTYPE OF (date_assignment);
    items : SET [1:?] OF date_assigned_item;
END_ENTITY; -- casting_date_assignment

ENTITY casting_document_reference
  SUBTYPE OF (document_reference);
    items : SET [1:?] OF document_assigned_item;
END_ENTITY; -- casting_document_reference

ENTITY casting_effectivity_assignment
  SUBTYPE OF (effectivity_assignment);
    items : SET [1:?] OF effectivity_assigned_item;
END_ENTITY; -- casting_effectivity_assignment

ENTITY casting_machining_allowance
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : SELF.description IN ['explicit surface allowance',
              'allowance along vector','allowance along normal'];
    wr2 : SIZEOF(QUERY ( sar <* USEDIN(SELF,
```

```
            'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
            'RELATING_SHAPE_ASPECT') | (sar.name =
            'geometry modification') )) = 1;
      wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
            SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'resultant geometry') )) = 1)) ))
            = 0;
      wr4 : (NOT (SELF.name = 'explicit surface allowance')) OR (SIZEOF(
            QUERY ( sar <* USEDIN(SELF,
            'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
            'RELATING_SHAPE_ASPECT') | (sar.name =
            'enveloping geometry') )) = 1);
      wr5 : (NOT (SELF.name IN ['allowance along vector',
            'allowance along normal'])) OR (SIZEOF(QUERY ( pd <*
            USEDIN(SELF,'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'feature parameters') )) = 1)) ))
            = 0);
      wr6 : (NOT (SELF.name = 'allowance along vector')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
            'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
            SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
             | (pdr.used_representation.name = 'feature parameters') )
             | (NOT ((2 <= SIZEOF(par_rep.used_representation.items))
            AND (SIZEOF(par_rep.used_representation.items) <= 4))) )) =
             0)) )) = 0);
      wr7 : (NOT (SELF.name = 'allowance along vector')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
            'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
            SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
             | (pdr.used_representation.name = 'feature parameters') )
             | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
            items | ('METAL_CASTING.VECTOR' IN TYPEOF(it)) )) = 1)) ))
            = 0)) )) = 0);
      wr8 : (NOT (SELF.name = 'allowance along vector')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
            'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
            SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
             | (pdr.used_representation.name = 'feature parameters') )
             | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
            items | ((SIZEOF([
            'METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
            'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
            'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 3)
            AND (it.name = 'offset amount') AND SIZEOF(it.qualifiers =
            1) AND SIZEOF(QUERY ( qual <* it.qualifiers | (NOT (
```

```
                'METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF(qual))) ) = 0)) ))
                <= 3)) )) = 0)) )) = 0);
    wr9 : (NOT (SELF.name = 'allowance along normal')) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
                'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
                SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
                 'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
                 | (pdr.used_representation.name = 'feature parameters') )
                 | (NOT ((1 <= SIZEOF(par_rep.used_representation.items))
                AND (SIZEOF(par_rep.used_representation.items) <= 3))) )) =
                 1)) )) = 0);
    wr10: (NOT (SELF.name = 'allowance along normal')) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
                'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
                SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
                 'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
                 | (pdr.used_representation.name = 'feature parameters') )
                 | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
                items | ((SIZEOF([
                'METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
                'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
                'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 3)
                AND (it.name = 'thickness') AND SIZEOF(it.qualifiers = 1)
                AND SIZEOF(QUERY ( qual <* it.qualifiers | (NOT (
                'METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF(qual))) ) = 0)) ))
                <= 3)) )) = 0)) )) = 0);
END_ENTITY; -- casting_machining_allowance

ENTITY casting_mold_design_element
  SUBTYPE OF (product_definition);
  WHERE
    wr1: SELF.frame_of_reference.name IN ['sand casting','die casting',
               'investment casting'];
    wr2: SIZEOF(QUERY ( acu <* QUERY ( pdr <* USEDIN(SELF,
               'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
               'RELATED_PRODUCT_DEFINITION') | (
               'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr)) )
                | (SIZEOF(['METAL_CASTING.SAND_MOLD_DEFINITION',
               'METAL_CASTING.DIE_MOLD_DEFINITION',
               'METAL_CASTING.INVESTMENT_MOLD_DEFINITION'] * TYPEOF(acu.
               relating.product_definition)) = 1) )) >= 1;
    wr3: SELF.frame_of_reference.life_cycle_stage = 'design';
    wr4: SELF.description IN ['die component','die sub-assembly','core',
               'flask','pattern and rigging assembly','pattern plate',
               'rigging component','insert','in-mold rigging component',
               'pattern','sand mold component','investment pattern',
               'sprue component'];
    wr5: (NOT (SELF.description IN ['pattern plate','rigging component',
               'pattern'])) OR (SIZEOF(QUERY ( acu <* QUERY ( pdr <*
               USEDIN(SELF,'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.'
               + 'RELATED_PRODUCT_DEFINITION') | (
               'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr)) )
```

```
             | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
            acu.relating_product_definition)) AND (acu.
            relating_product_definition.description =
            'pattern and rigging assembly')) )) >= 1);
    wr6: (NOT (SELF.description = 'sand mold component')) OR (SIZEOF(
            QUERY ( pdu <* QUERY ( pdr <* USEDIN(SELF,
            'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
            'RELATING_PRODUCT_DEFINITION') | (
            'METAL_CASTING.PRODUCT_DEFINITION_USAGE' IN TYPEOF(pdr)) )
             | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
            pdu.relating_product_definition)) AND (pdu.
            relating_product_definition.description = 'flask')) )) = 1);
    wr7: (NOT (SELF.description = 'sand mold component')) OR (SIZEOF(
            QUERY ( pdu <* QUERY ( pdr <* USEDIN(SELF,
            'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
            'RELATING_PRODUCT_DEFINITION') | (
            'METAL_CASTING.PRODUCT_DEFINITION_USAGE' IN TYPEOF(pdr)) )
             | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
            pdu.relating_product_definition)) AND (pdu.
            relating_product_definition.description =
            'pattern and rigging assembly')) )) = 1);
    wr8: (NOT (SELF.description = 'pattern and rigging assembly')) OR (
            SIZEOF(QUERY ( pdu <* QUERY ( pdr <* USEDIN(SELF,
            'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
            'RELATING_PRODUCT_DEFINITION') | (
            'METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr)) )
             | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
            pdu.relating_product_definition)) AND (pdu.
            relating_product_definition.description = 'pattern plate')) ))
            = 1);
    wr9: (NOT (SELF.description = 'investment pattern')) OR (SIZEOF(
            QUERY ( pdu <* QUERY ( pdr <* USEDIN(SELF,
            'METAL_CASTING.PRODUCT_DEFINITION_RELATIONSHIP.' +
            'RELATING_PRODUCT_DEFINITION') | (
            'METAL_CASTING.PRODUCT_DEFINITION_USAGE' IN TYPEOF(pdr)) )
             | (('METAL_CASTING.DIE_MOLD_DEFINITION' IN TYPEOF(pdu.
            related_product_definition)) AND (pdu.
            relating_product_definition.frame_of_reference.name =
            'investment casting')) )) = 1);
END_ENTITY; -- casting_mold_design_element

ENTITY casting_organization_assignment
  SUBTYPE OF (organization_assignment);
    items : SET [1:?] OF organization_assigned_item;
END_ENTITY; -- casting_organization_assignment

ENTITY casting_part_definition
  SUBTYPE OF (product_definition);
  WHERE
    wr1: SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
            'METAL_CASTING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')
             | (NOT (prpc.name = 'casting')) )) = 0;
```

```
    wr2: SELF.frame_of_reference.frame_of_reference.application =
              'casting';
    wr3: SELF.frame_of_reference.name IN ['foundry part','customer part',
              'raw','finished'];
    wr4: SELF.frame_of_reference.life_cycle_stage IN ['design',
              'production'];
    wr5: (NOT (SELF.frame_of_reference.life_cycle_stage = 'production'))
              OR (SELF.frame_of_reference.name IN ['raw','finished']);
    wr6: (NOT (SELF.description = 'artifact')) OR ((SIZEOF(USEDIN(SELF,
              'METAL_CASTING.CASTING_DATE_ASSIGNMENT.' + 'ITEMS')) +
              SIZEOF(USEDIN(SELF,'METAL_CASTING.' +
              'CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS'))) = 1);
END_ENTITY; -- casting_part_definition

ENTITY casting_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
    items : SET [1:?] OF person_and_organization_assigned_item;
END_ENTITY; -- casting_person_and_organization_assignment

ENTITY casting_round_corner_transition
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
              'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
              'RELATING_SHAPE_ASPECT') | (sar.name =
              'geometry modification') )) = 1;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
              SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
              used_representation.name = 'resultant geometry') )) = 1)) ))
              = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
              SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
              used_representation.name = 'feature parameters') )) = 1)) ))
              = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
              SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
               'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
               | (pdr.used_representation.name = 'feature parameters') )
               | (NOT ((1 <= SIZEOF(par_rep.used_representation.items))
              AND (SIZEOF(par_rep.used_representation.items) <= 3))) )) =
              0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
              SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
               'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
               | (pdr.used_representation.name = 'feature parameters') )
               | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
```

```
            items | ((SIZEOF([
            'METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
            'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
            'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 3)
            AND (it.name = 'radius') AND SIZEOF(it.qualifiers = 1) AND (
            SIZEOF(QUERY ( qual <* it.qualifiers | (NOT (
            'METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF(qual))) )) = 0)) ))
            <= 3)) )) = 0)) )) = 0;
END_ENTITY; -- casting_round_corner_transition

ENTITY casting_round_edge_transition
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
            'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
            'RELATING_SHAPE_ASPECT') | (sar.name =
            'geometry modification') )) = 1;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
            SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'resultant geometry') )) = 1)) ))
            = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
            SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'feature parameters') )) = 1)) ))
            = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
            SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
             'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
             | (pdr.used_representation.name = 'feature parameters') )
             | (NOT ((1 <= SIZEOF(par_rep.used_representation.items))
            AND (SIZEOF(par_rep.used_representation.items) <= 3))) )) =
            0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
            SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
             'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
             | (pdr.used_representation.name = 'feature parameters') )
             | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
            items | ((SIZEOF([
            'METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
            'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
            'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 3)
            AND (it.name = 'radius') AND SIZEOF(it.qualifiers = 1) AND (
            SIZEOF(QUERY ( qual <* it.qualifiers | (NOT (
            'METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF(qual))) )) = 0)) ))
            <= 3)) )) = 0)) )) = 0;
END_ENTITY; -- casting_round_edge_transition
```

```
ENTITY casting_taper
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
             'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
             'RELATING_SHAPE_ASPECT') | (sar.name =
             'geometry modification') )) = 1;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
             used_representation.name = 'resultant geometry') )) = 1)) ))
             = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
             used_representation.name = 'feature parameters') )) = 1)) ))
             = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
              'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
              | (pdr.used_representation.name = 'feature parameters') )
              | (NOT ((1 <= SIZEOF(par_rep.used_representation.items))
             AND (SIZEOF(par_rep.used_representation.items) <= 7))) )) =
             0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
              'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
              | (pdr.used_representation.name = 'feature parameters') )
              | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
             items | ((SIZEOF([
             'METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
             'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
             'METAL_CASTING.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(it))
             = 3) AND (it.name = 'angle') AND SIZEOF(it.qualifiers = 1)
             AND (SIZEOF(QUERY ( qual <* it.qualifiers | (NOT (
             'METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF(qual))) )) = 0)) ))
             <= 3)) )) = 0)) )) = 0;
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
              'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
              | (pdr.used_representation.name = 'feature parameters') )
              | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
             items | ((SIZEOF([
             'METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
             'METAL_CASTING.QUALIFIED_REPRESENTATION_ITEM',
             'METAL_CASTING.RATIO_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 3)
```

```
              AND (it.name = 'ratio') AND SIZEOF(it.qualifiers = 1) AND (
              SIZEOF(QUERY ( qual <* it.qualifiers | (NOT (
              'METAL_CASTING.TYPE_QUALIFIER' IN TYPEOF(qual))) )) = 0)) ))
              <= 3)) )) = 0)) )) = 0;
      wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
              SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
               'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
               | (pdr.used_representation.name = 'feature parameters') )
               | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
              items | (('METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN
               TYPEOF(it)) AND (it.name = 'taper method') AND (it.
              description IN ['plus','minus','average'])) )) = 1)) )) = 0)) ))
              = 0;
      wr8: SIZEOF(QUERY ( sar <* USEDIN(SELF,
               'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
               | (sar.related_shape_aspect.description = 'parting surface') ))
              = 1;
      wr9: SIZEOF(QUERY ( sar <* USEDIN(SELF,
               'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
               | (sar.related_shape_aspect.description = 'reference edge') ))
              = 1;
  END_ENTITY; -- casting_taper

  ENTITY change_from_assignment
    SUBTYPE OF (action_assignment);
      items : SET [1:?] OF change_item;
  END_ENTITY; -- change_from_assignment

  ENTITY change_to_assignment
    SUBTYPE OF (action_assignment);
      items : SET [1:?] OF change_item;
  END_ENTITY; -- change_to_assignment

  ENTITY characterized_object;
      name        : label;
      description : text;
  END_ENTITY; -- characterized_object

  ENTITY chemical_structure_element
    SUBTYPE OF (product_definition);
    WHERE
      wr1: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
               | ((
              'METAL_CASTING.PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP' IN
               TYPEOF(pdr)) AND (SIZEOF(QUERY ( prpc <* USEDIN(pdr.
              relating_product_definition.formation.of_product,
              'METAL_CASTING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')
               | (prpc.name = 'material') )) = 1)) )) >= 1;
  END_ENTITY; -- chemical_structure_element
```

```
ENTITY circle
  SUBTYPE OF (conic);
    radius : positive_length_measure;
END_ENTITY; -- circle

ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SELF\geometric_tolerance.name = 'circular runout';
    wr2: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
             datum_system) <= 2;
END_ENTITY; -- circular_runout_tolerance

ENTITY closed_shell
  SUBTYPE OF (connected_face_set);
END_ENTITY; -- closed_shell

ENTITY composite_curve
  SUBTYPE OF (bounded_curve);
    segments       : LIST [1:?] OF composite_curve_segment;
    self_intersect : LOGICAL;
  DERIVE
    n_segments   : INTEGER := SIZEOF(segments);
    closed_curve : LOGICAL := segments[n_segments].transition <>
                     discontinuous;
  WHERE
    wr1: ((NOT closed_curve) AND (SIZEOF(QUERY ( temp <* segments | (
             temp.transition = discontinuous) )) = 1)) OR (closed_curve
             AND (SIZEOF(QUERY ( temp <* segments | (temp.transition =
             discontinuous) )) = 0));
END_ENTITY; -- composite_curve

ENTITY composite_curve_on_surface
  SUBTYPE OF (composite_curve);
  DERIVE
    basis_surface : SET [0:2] OF surface := get_basis_surface(SELF);
  WHERE
    wr1: SIZEOF(basis_surface) > 0;
    wr2: constraints_composite_curve_on_surface(SELF);
END_ENTITY; -- composite_curve_on_surface

ENTITY composite_curve_segment;
    transition   : transition_code;
    same_sense   : BOOLEAN;
    parent_curve : curve;
  INVERSE
    using_curves : BAG [1:?] OF composite_curve FOR segments;
  WHERE
    wr1: 'METAL_CASTING.BOUNDED_CURVE' IN TYPEOF(parent_curve);
END_ENTITY; -- composite_curve_segment

ENTITY concentricity_tolerance
```

```
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SELF\geometric_tolerance.name = 'concentricity';
    wr2: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
             datum_system) = 1;
END_ENTITY; -- concentricity_tolerance

ENTITY concurrent_action_method
  SUBTYPE OF (action_method_relationship);
END_ENTITY; -- concurrent_action_method

ENTITY conic
  SUPERTYPE OF (ONEOF (circle,ellipse,hyperbola,parabola))
  SUBTYPE OF (curve);
    position : axis2_placement;
END_ENTITY; -- conic

ENTITY conical_surface
  SUBTYPE OF (elementary_surface);
    radius     : length_measure;
    semi_angle : plane_angle_measure;
  WHERE
    wr1: radius >= 0;
END_ENTITY; -- conical_surface

ENTITY connected_face_set
  SUPERTYPE OF (ONEOF (closed_shell,open_shell))
  SUBTYPE OF (topological_representation_item);
    cfs_faces : SET [1:?] OF face;
END_ENTITY; -- connected_face_set

ENTITY coordinated_universal_time_offset;
    hour_offset   : hour_in_day;
    minute_offset : OPTIONAL minute_in_hour;
    sense         : ahead_or_behind;
END_ENTITY; -- coordinated_universal_time_offset

ENTITY curve
  SUPERTYPE OF (ONEOF (line,conic,pcurve,surface_curve,offset_curve_2d))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- curve

ENTITY cylindrical_surface
  SUBTYPE OF (elementary_surface);
    radius : positive_length_measure;
END_ENTITY; -- cylindrical_surface

ENTITY data_curve_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: SIZEOF(SELF.items) = 2;
    wr2: SIZEOF(QUERY ( it <* SELF.items | ((
```

```
              'METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
              AND (it.name = 'interpolation method')) )) = 1;
     wr3: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
              'METAL_CASTING.CARTESIAN_POINT','METAL_CASTING.POLYLINE'] *
              TYPEOF(it)) = 1) )) = 1;
     wr4: 'METAL_CASTING.GLOBAL_UNIT_ASSIGNED_CONTEXT' IN TYPEOF(SELF.
              context_of_items);
     wr5: SIZEOF(QUERY ( gri <* QUERY ( it <* SELF.items | (
              'METAL_CASTING.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF(it)) )
               | (NOT (gri.dim = 2)) )) = 0;
     wr6: SIZEOF(SELF.context_of_items\global_unit_assigned_context.units)
              = 1;
     wr7: SIZEOF(QUERY ( unit <* SELF.context_of_items\
              global_unit_assigned_context.units | (NOT ((
              'METAL_CASTING.DERIVED_UNIT' IN TYPEOF(unit)) AND (SIZEOF(
              unit.elements) = 2))) )) = 0;
END_ENTITY; -- data_curve_representation

ENTITY date
  SUPERTYPE OF (calendar_date);
    year_component : year_number;
END_ENTITY; -- date

ENTITY date_and_time;
    date_component : date;
    time_component : local_time;
END_ENTITY; -- date_and_time

ENTITY date_and_time_assignment
  ABSTRACT SUPERTYPE;
    assigned_date_and_time : date_and_time;
    role                   : date_time_role;
END_ENTITY; -- date_and_time_assignment

ENTITY date_assignment
  ABSTRACT SUPERTYPE;
    assigned_date : date;
    role          : date_role;
END_ENTITY; -- date_assignment

ENTITY date_role;
    name : label;
END_ENTITY; -- date_role

ENTITY date_time_role;
    name : label;
END_ENTITY; -- date_time_role

ENTITY datum
  SUBTYPE OF (shape_aspect);
    identification : identifier;
  INVERSE
```

```
    established_by_relationships : SET [1:?] OF
                                       shape_aspect_relationship FOR
                                      related_shape_aspect;
  WHERE
    wr1: SIZEOF(QUERY ( x <* SELF.established_by_relationships | (
             SIZEOF(TYPEOF(x.relating_shape_aspect) * [
             'METAL_CASTING.DATUM_FEATURE',
             'METAL_CASTING.DATUM_TARGET_FEATURE']) <> 1) )) = 0;
END_ENTITY; -- datum

ENTITY datum_reference;
    precedence       : INTEGER;
    referenced_datum : datum;
  WHERE
    wr1: precedence > 0;
END_ENTITY; -- datum_reference

ENTITY definitional_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: 'METAL_CASTING.PARAMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF(
             SELF\representation.context_of_items);
END_ENTITY; -- definitional_representation

ENTITY degenerate_toroidal_surface
  SUBTYPE OF (toroidal_surface);
    select_outer : BOOLEAN;
  WHERE
    wr1: major_radius < minor_radius;
END_ENTITY; -- degenerate_toroidal_surface

ENTITY derived_unit;
    elements : SET [1:?] OF derived_unit_element;
  WHERE
    wr1: (SIZEOF(elements) > 1) OR ((SIZEOF(elements) = 1) AND (elements
             [1].exponent <> 1));
END_ENTITY; -- derived_unit

ENTITY derived_unit_element;
    unit     : named_unit;
    exponent : REAL;
END_ENTITY; -- derived_unit_element

ENTITY descriptive_representation_item
  SUBTYPE OF (representation_item);
    description : text;
END_ENTITY; -- descriptive_representation_item

ENTITY die_mold_definition
  SUBTYPE OF (product_definition);
  WHERE
    wr1: SELF.frame_of_reference.name IN ['die casting',
```

```
                 'investment casting'];
        wr2: SELF.frame_of_reference.frame_of_reference.application =
                 'casting';
        wr3: SELF.frame_of_reference.life_cycle_stage = 'design';
    END_ENTITY; -- die_mold_definition

    ENTITY dimensional_exponents;
        length_exponent                    : REAL;
        mass_exponent                      : REAL;
        time_exponent                      : REAL;
        electric_current_exponent          : REAL;
        thermodynamic_temperature_exponent : REAL;
        amount_of_substance_exponent       : REAL;
        luminous_intensity_exponent        : REAL;
    END_ENTITY; -- dimensional_exponents

    ENTITY dimensional_location
      SUBTYPE OF (shape_aspect_relationship);
    END_ENTITY; -- dimensional_location

    ENTITY dimensional_size;
        applies_to : shape_aspect;
        name       : label;
      WHERE
        wr1: applies_to.product_definitional = TRUE;
    END_ENTITY; -- dimensional_size

    ENTITY directed_action
      SUBTYPE OF (executed_action);
        directive : action_directive;
    END_ENTITY; -- directed_action

    ENTITY direction
      SUBTYPE OF (geometric_representation_item);
        direction_ratios : LIST [2:3] OF REAL;
      WHERE
        wr1: SIZEOF(QUERY ( tmp <* direction_ratios | (tmp <> 0) )) > 0;
    END_ENTITY; -- direction

    ENTITY document;
        id          : identifier;
        name        : label;
        description : text;
        kind        : document_type;
      UNIQUE
        ur1 : id;
    END_ENTITY; -- document

    ENTITY document_reference
      ABSTRACT SUPERTYPE;
        assigned_document : document;
        source            : label;
```

```
END_ENTITY; -- document_reference

ENTITY document_type;
    product_data_type : label;
END_ENTITY; -- document_type

ENTITY document_usage_constraint;
    source                : document;
    subject_element       : label;
    subject_element_value : text;
END_ENTITY; -- document_usage_constraint

ENTITY edge
  SUPERTYPE OF (ONEOF (edge_curve,oriented_edge))
  SUBTYPE OF (topological_representation_item);
    edge_start : vertex;
    edge_end   : vertex;
END_ENTITY; -- edge

ENTITY edge_curve
  SUBTYPE OF (edge, geometric_representation_item);
    edge_geometry : curve;
    same_sense    : BOOLEAN;
END_ENTITY; -- edge_curve

ENTITY edge_loop
  SUBTYPE OF (loop, path);
  DERIVE
    ne : INTEGER := SIZEOF(SELF\path.edge_list);
  WHERE
    wr1: SELF\path.edge_list[1].edge_start :=: SELF\path.edge_list[ne].
             edge_end;
END_ENTITY; -- edge_loop

ENTITY effectivity
  SUPERTYPE OF (lot_effectivity);
    id : identifier;
END_ENTITY; -- effectivity

ENTITY effectivity_assignment
  ABSTRACT SUPERTYPE;
    assigned_effectivity : effectivity;
END_ENTITY; -- effectivity_assignment

ENTITY elementary_brep_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1 : SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
              'METAL_CASTING.MANIFOLD_SOLID_BREP',
              'METAL_CASTING.FACETED_BREP','METAL_CASTING.MAPPED_ITEM',
              'METAL_CASTING.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1)) ))
              = 0;
```

```
wr2 : SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
        'METAL_CASTING.MANIFOLD_SOLID_BREP',
        'METAL_CASTING.MAPPED_ITEM'] * TYPEOF(it)) = 1) )) > 0;
wr3 : SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
        'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
        NOT (SIZEOF(QUERY ( csh <* msb_shells(msb,
        'AIC_ELEMENTARY_BREP') | (NOT (SIZEOF(QUERY ( fcs <* csh.
        cfs_faces | (NOT ('METAL_CASTING.FACE_SURFACE' IN TYPEOF(
        fcs))) )) = 0)) )) = 0)) )) = 0;
wr4 : SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
        'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
        NOT (SIZEOF(QUERY ( csh <* msb_shells(msb,
        'AIC_ELEMENTARY_BREP') | (NOT (SIZEOF(QUERY ( fcs <* csh\
        connected_face_set.cfs_faces | (NOT (
        'METAL_CASTING.ELEMENTARY_SURFACE' IN TYPEOF(fcs\
        face_surface.face_geometry))) )) = 0)) )) = 0)) )) = 0;
wr5 : SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
        'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
        NOT (SIZEOF(QUERY ( csh <* msb_shells(msb,
        'AIC_ELEMENTARY_BREP') | (NOT (SIZEOF(QUERY ( fcs <* csh\
        connected_face_set.cfs_faces | (NOT (SIZEOF(
        QUERY ( elp_fbnds <* QUERY ( bnds <* fcs.bounds | (
        'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (NOT (
        SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list | (NOT
        ('METAL_CASTING.EDGE_CURVE' IN TYPEOF(oe.edge_element))) ))
        = 0)) )) = 0)) )) = 0)) )) = 0)) )) = 0;
wr6 : SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
        'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
        NOT (SIZEOF(QUERY ( csh <* msb_shells(msb,
        'AIC_ELEMENTARY_BREP') | (NOT (SIZEOF(QUERY ( fcs <* csh\
        connected_face_set.cfs_faces | (NOT (SIZEOF(
        QUERY ( elp_fbnds <* QUERY ( bnds <* fcs.bounds | (
        'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (NOT (
        SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list | (NOT
        (SIZEOF(['METAL_CASTING.LINE','METAL_CASTING.CONIC',
        'METAL_CASTING.POLYLINE'] * TYPEOF(oe.edge_element\
        edge_curve.edge_geometry)) = 1)) )) = 0)) )) = 0)) )) = 0)) ))
        = 0)) )) = 0;
wr7 : SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
        'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
        NOT (SIZEOF(QUERY ( csh <* msb_shells(msb,
        'AIC_ELEMENTARY_BREP') | (NOT (SIZEOF(QUERY ( fcs <* csh\
        connected_face_set.cfs_faces | (NOT (SIZEOF(
        QUERY ( elp_fbnds <* QUERY ( bnds <* fcs.bounds | (
        'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (NOT (
        SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list | (NOT
        (('METAL_CASTING.VERTEX_POINT' IN TYPEOF(oe.edge_start))
        AND ('METAL_CASTING.VERTEX_POINT' IN TYPEOF(oe.edge_end)))) ))
        = 0)) )) = 0)) )) = 0)) )) = 0)) )) = 0;
wr8 : SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
        'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
        NOT (SIZEOF(QUERY ( csh <* msb_shells(msb,
```

```
            'AIC_ELEMENTARY_BREP') | (NOT (SIZEOF(QUERY ( fcs <* csh\
            connected_face_set.cfs_faces | (NOT (SIZEOF(
            QUERY ( elp_fbnds <* QUERY ( bnds <* fcs.bounds | (
            'METAL_CASTING.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (
            'METAL_CASTING.ORIENTED_PATH' IN TYPEOF(elp_fbnds.bound)) ))
            = 0)) )) = 0)) )) = 0)) )) = 0;
    wr9 : SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
            'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
            'METAL_CASTING.ORIENTED_CLOSED_SHELL' IN TYPEOF(msb\
            manifold_solid_brep.outer)) )) = 0;
    wr10: SIZEOF(QUERY ( brv <* QUERY ( it <* SELF.items | (
            'METAL_CASTING.BREP_WITH_VOIDS' IN TYPEOF(it)) ) | (NOT (
            SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids | csh\
            oriented_closed_shell.orientation )) = 0)) )) = 0;
    wr11: SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
            'METAL_CASTING.MAPPED_ITEM' IN TYPEOF(it)) ) | (NOT (
            'METAL_CASTING.ELEMENTARY_BREP_SHAPE_REPRESENTATION' IN
            TYPEOF(mi\mapped_item.mapping_source.mapped_representation))) ))
            = 0;
    wr12: SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
            'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
            NOT (SIZEOF(QUERY ( csh <* msb_shells(msb,
            'AIC_ELEMENTARY_BREP') | (NOT (SIZEOF(QUERY ( fcs <* csh\
            connected_face_set.cfs_faces | (NOT (SIZEOF(
            QUERY ( vlp_fbnds <* QUERY ( bnds <* fcs.bounds | (
            'METAL_CASTING.VERTEX_LOOP' IN TYPEOF(bnds.bound)) ) | (
            NOT (('METAL_CASTING.VERTEX_POINT' IN TYPEOF(vlp_fbnds\
            face_bound.bound\vertex_loop.loop_vertex)) AND (
            'METAL_CASTING.CARTESIAN_POINT' IN TYPEOF(vlp_fbnds\
            face_bound.bound\vertex_loop.loop_vertex\vertex_point.
            vertex_geometry)))) )) = 0)) )) = 0)) )) = 0)) )) = 0;
END_ENTITY; -- elementary_brep_shape_representation

ENTITY elementary_surface
  SUPERTYPE OF (ONEOF (plane,cylindrical_surface,conical_surface,
      spherical_surface,toroidal_surface))
  SUBTYPE OF (surface);
    position : axis2_placement_3d;
END_ENTITY; -- elementary_surface

ENTITY ellipse
  SUBTYPE OF (conic);
    semi_axis_1 : positive_length_measure;
    semi_axis_2 : positive_length_measure;
END_ENTITY; -- ellipse

ENTITY executed_action
  SUBTYPE OF (action);
END_ENTITY; -- executed_action

ENTITY face
  SUPERTYPE OF (ONEOF (face_surface,oriented_face))
```

```
  SUBTYPE OF (topological_representation_item);
    bounds : SET [1:?] OF face_bound;
  WHERE
    wr1: NOT mixed_loop_type_set(list_to_set(list_face_loops(SELF)));
    wr2: SIZEOF(QUERY ( temp <* bounds | (
             'METAL_CASTING.FACE_OUTER_BOUND' IN TYPEOF(temp)) )) <= 1;
END_ENTITY; -- face

ENTITY face_bound
  SUBTYPE OF (topological_representation_item);
    bound       : loop;
    orientation : BOOLEAN;
END_ENTITY; -- face_bound

ENTITY face_outer_bound
  SUBTYPE OF (face_bound);
END_ENTITY; -- face_outer_bound

ENTITY face_surface
  SUBTYPE OF (face, geometric_representation_item);
    face_geometry : surface;
    same_sense    : BOOLEAN;
END_ENTITY; -- face_surface

ENTITY faceted_brep
  SUBTYPE OF (manifold_solid_brep);
END_ENTITY; -- faceted_brep

ENTITY faceted_brep_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
             'METAL_CASTING.FACETED_BREP','METAL_CASTING.MAPPED_ITEM',
             'METAL_CASTING.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1)) )) =
              0;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
             'METAL_CASTING.FACETED_BREP','METAL_CASTING.MAPPED_ITEM'] *
             TYPEOF(it)) = 1) )) > 0;
    wr3: SIZEOF(QUERY ( fbrep <* QUERY ( it <* SELF.items | (
             'METAL_CASTING.FACETED_BREP' IN TYPEOF(it)) ) | (NOT (
             SIZEOF(QUERY ( csh <* msb_shells(fbrep,'AIC_FACETED_BREP')
              | (NOT (SIZEOF(QUERY ( fcs <* csh.cfs_faces | (NOT ((
             'METAL_CASTING.FACE_SURFACE' IN TYPEOF(fcs)) AND (
             'METAL_CASTING.PLANE' IN TYPEOF(fcs\face_surface.
             face_geometry)) AND ('METAL_CASTING.CARTESIAN_POINT' IN
             TYPEOF(fcs\face_surface.face_geometry\elementary_surface.
             position.location)))) )) = 0)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( fbrep <* QUERY ( it <* SELF.items | (
             'METAL_CASTING.FACETED_BREP' IN TYPEOF(it)) ) | (NOT (
             SIZEOF(QUERY ( csh <* msb_shells(fbrep,'AIC_FACETED_BREP')
              | (NOT (SIZEOF(QUERY ( fcs <* csh.cfs_faces | (NOT (SIZEOF(
             QUERY ( bnds <* fcs.bounds | (
```

```
            'METAL_CASTING.FACE_OUTER_BOUND' IN TYPEOF(bnds)) )) = 1)) ))
            = 0)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
            'METAL_CASTING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
            'METAL_CASTING.ORIENTED_CLOSED_SHELL' IN TYPEOF(msb\
            manifold_solid_brep.outer)) )) = 0;
    wr6: SIZEOF(QUERY ( brv <* QUERY ( it <* SELF.items | (
            'METAL_CASTING.BREP_WITH_VOIDS' IN TYPEOF(it)) ) | (NOT (
            SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids | csh\
            oriented_closed_shell.orientation )) = 0)) )) = 0;
    wr7: SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
            'METAL_CASTING.MAPPED_ITEM' IN TYPEOF(it)) ) | (NOT (
            'METAL_CASTING.FACETED_BREP_SHAPE_REPRESENTATION' IN TYPEOF(
            mi\mapped_item.mapping_source.mapped_representation))) )) =
            0;
END_ENTITY; -- faceted_brep_shape_representation

ENTITY functionally_defined_transformation;
    name        : label;
    description : text;
END_ENTITY; -- functionally_defined_transformation

ENTITY geometric_curve_set
  SUBTYPE OF (geometric_set);
  WHERE
    wr1: SIZEOF(QUERY ( temp <* SELF\geometric_set.elements | (
            'METAL_CASTING.SURFACE' IN TYPEOF(temp)) )) = 0;
END_ENTITY; -- geometric_curve_set

ENTITY geometric_representation_context
  SUBTYPE OF (representation_context);
    coordinate_space_dimension : dimension_count;
END_ENTITY; -- geometric_representation_context

ENTITY geometric_representation_item
  SUPERTYPE OF (ONEOF (point,direction,vector,placement,
      cartesian_transformation_operator,curve,surface,edge_curve,
      face_surface,poly_loop,vertex_point,solid_model,geometric_set))
  SUBTYPE OF (representation_item);
  DERIVE
    dim : dimension_count := dimension_of(SELF);
  WHERE
    wr1: SIZEOF(QUERY ( using_rep <* using_representations(SELF) | (NOT
            ('METAL_CASTING.GEOMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF(
            using_rep.context_of_items))) )) = 0;
END_ENTITY; -- geometric_representation_item

ENTITY geometric_set
  SUPERTYPE OF (geometric_curve_set)
  SUBTYPE OF (geometric_representation_item);
    elements : SET [1:?] OF geometric_set_select;
END_ENTITY; -- geometric_set
```

```
ENTITY geometric_tolerance;
    name                    : label;
    description             : text;
    magnitude               : measure_with_unit;
    toleranced_shape_aspect : shape_aspect;
  WHERE
    wr1: magnitude.value_component >= 0;
END_ENTITY; -- geometric_tolerance

ENTITY geometric_tolerance_with_datum_reference
  SUBTYPE OF (geometric_tolerance);
    datum_system : SET [1:?] OF datum_reference;
END_ENTITY; -- geometric_tolerance_with_datum_reference

ENTITY geometric_tolerance_with_defined_unit
  SUBTYPE OF (geometric_tolerance);
    unit_size : measure_with_unit;
  WHERE
    wr1: unit_size.value_component > 0;
END_ENTITY; -- geometric_tolerance_with_defined_unit

ENTITY geometrically_bounded_2d_wireframe_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SELF.context_of_items\geometric_representation_context.
             coordinate_space_dimension = 2;
    wr2: SIZEOF(QUERY ( item <* SELF.items | (NOT (SIZEOF(TYPEOF(item) *
              ['METAL_CASTING.GEOMETRIC_CURVE_SET',
             'METAL_CASTING.AXIS2_PLACEMENT_2D',
             'METAL_CASTING.MAPPED_ITEM']) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( item <* SELF.items | (SIZEOF(TYPEOF(item) * [
             'METAL_CASTING.GEOMETRIC_CURVE_SET',
             'METAL_CASTING.MAPPED_ITEM']) = 1) )) >= 1;
    wr4: SIZEOF(QUERY ( mi <* QUERY ( item <* SELF.items | (
             'METAL_CASTING.MAPPED_ITEM' IN TYPEOF(item)) ) | (NOT ((
             'METAL_CASTING.' +
             'GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION') IN
             TYPEOF(mi\mapped_item.mapping_source.mapped_representation))) ))
             = 0;
    wr5: SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
             'METAL_CASTING.GEOMETRIC_CURVE_SET' IN TYPEOF(item)) ) | (
             NOT (SIZEOF(QUERY ( elem <* gcs\geometric_set.elements | (
             NOT (SIZEOF(TYPEOF(elem) * ['METAL_CASTING.B_SPLINE_CURVE',
             'METAL_CASTING.CIRCLE','METAL_CASTING.COMPOSITE_CURVE',
             'METAL_CASTING.ELLIPSE','METAL_CASTING.OFFSET_CURVE_2D',
             'METAL_CASTING.POINT','METAL_CASTING.POLYLINE',
             'METAL_CASTING.TRIMMED_CURVE']) = 1)) )) = 0)) )) = 0;
    wr6: SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
             'METAL_CASTING.GEOMETRIC_CURVE_SET' IN TYPEOF(item)) ) | (
             NOT (SIZEOF(QUERY ( crv <* QUERY ( elem <* gcs\geometric_set
             .elements | ('METAL_CASTING.CURVE' IN TYPEOF(elem)) ) | (
```

```
            NOT valid_basis_curve_in_2d_wireframe(crv,
            'AIC_GEOMETRICALLY_BOUNDED_2D_WIRERAME')) )) = 0)) )) = 0;
    wr7: SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
            'METAL_CASTING.GEOMETRIC_CURVE_SET' IN TYPEOF(item)) ) | (
            NOT (SIZEOF(QUERY ( pnt <* QUERY ( elem <* gcs\geometric_set
            .elements | ('METAL_CASTING.POINT' IN TYPEOF(elem)) ) | (
            NOT (SIZEOF(TYPEOF(pnt) * ['METAL_CASTING.CARTESIAN_POINT',
            'METAL_CASTING.POINT_ON_CURVE']) = 1)) )) = 0)) )) = 0;
    wr8: SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
            'METAL_CASTING.GEOMETRIC_CURVE_SET' IN TYPEOF(item)) ) | (
            NOT (SIZEOF(QUERY ( pl <* QUERY ( elem <* gcs\geometric_set.
            elements | ('METAL_CASTING.POLYLINE' IN TYPEOF(elem)) ) | (
            NOT (SIZEOF(pl\polyline.points) > 2)) )) = 0)) )) = 0;
END_ENTITY; -- geometrically_bounded_2d_wireframe_representation

ENTITY global_unit_assigned_context
  SUBTYPE OF (representation_context);
    units : SET [1:?] OF unit;
END_ENTITY; -- global_unit_assigned_context

ENTITY heat_effectivity
  SUBTYPE OF (effectivity);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,
            'METAL_CASTING.CASTING_DATE_ASSIGNMENT.ITEMS')) + SIZEOF(
            USEDIN(SELF,'METAL_CASTING.' +
            'CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS'))) = 1;
    wr2: (NOT (SIZEOF(USEDIN(SELF,
            'METAL_CASTING.CASTING_DATE_ASSIGNMENT.ITEMS')) = 1)) OR (
            SIZEOF(QUERY ( cda <* USEDIN(SELF,
            'METAL_CASTING.CASTING_DATE_ASSIGNMENT.ITEMS') | (NOT (cda.
            role.name = 'production date')) )) = 0);
    wr3: (NOT (SIZEOF(USEDIN(SELF,
            'METAL_CASTING.CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS')) = 1))
            OR (SIZEOF(QUERY ( cdta <* USEDIN(SELF,
            'METAL_CASTING.CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS') | (
            NOT (cdta.role.name = 'production date')) )) = 0);
    wr4: (NOT (SIZEOF(USEDIN(SELF,
            'METAL_CASTING.EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY'))
            >= 1)) OR (SIZEOF(QUERY ( cea <* QUERY ( ea <* USEDIN(SELF,
            'METAL_CASTING.EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY')
             | ('METAL_CASTING.CASTING_EFFECTIVITY_ASSIGNMENT' IN
            TYPEOF(ea)) ) | (NOT (SIZEOF(QUERY ( it <* cea.items | (NOT
            (('METAL_CASTING.CASTING_PART_DEFINITION' IN TYPEOF(it)) AND
             (it.description = 'artifact'))) )) = 0)) )) = 0);
END_ENTITY; -- heat_effectivity

ENTITY hyperbola
  SUBTYPE OF (conic);
    semi_axis      : positive_length_measure;
    semi_imag_axis : positive_length_measure;
END_ENTITY; -- hyperbola
```

```
ENTITY identification_marking
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
             'METAL_CASTING.SHAPE_ASPECT_RELATIONSHIP.' +
             'RELATING_SHAPE_ASPECT') | (sar.name =
             'geometry modification') )) = 1;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
             used_representation.name = 'resultant geometry') )) = 1)) ))
             = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( pdr <* USEDIN(pd,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
             used_representation.name = 'feature parameters') )) = 1)) ))
             = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
              'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
              | (pdr.used_representation.name = 'feature parameters') )
              | (NOT (SIZEOF(par_rep.used_representation.items) = 2)) ))
             = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
              'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
              | (pdr.used_representation.name = 'feature parameters') )
              | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
             items | ((SIZEOF([
             'METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
             'METAL_CASTING.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2)
             AND (it.name = 'height')) )) = 1)) )) = 0)) )) = 0;
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( par_rep <* QUERY ( pdr <* USEDIN(pd,
              'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
              | (pdr.used_representation.name = 'feature parameters') )
              | (NOT (SIZEOF(QUERY ( it <* par_rep.used_representation.
             items | (('METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN
              TYPEOF(it)) AND (it.name = 'raised') AND (it.description IN
              ['raised','not raised'])) )) = 1)) )) = 0)) )) = 0;
    wr7: SIZEOF(QUERY ( cdr <* USEDIN(SELF,
             'METAL_CASTING.CASTING_DOCUMENT_REFERENCE.ITEMS') | (cdr.
             assigned_document.kind.product_data_type = 'graphics') )) =
             1;
END_ENTITY; -- identification_marking
```

```
ENTITY investment_mold_definition
  SUBTYPE OF (product_definition);
  WHERE
    wr1: SELF.frame_of_reference.name = 'investment casting';
    wr2: SELF.frame_of_reference.frame_of_reference.application =
             'casting';
    wr3: SELF.frame_of_reference.life_cycle_stage = 'design';
END_ENTITY; -- investment_mold_definition

ENTITY investment_mold_pattern_definition
  SUBTYPE OF (product_definition);
  WHERE
    wr1: SELF.frame_of_reference.name = 'investment casting';
    wr2: SELF.frame_of_reference.frame_of_reference.application =
             'casting';
    wr3: SELF.frame_of_reference.life_cycle_stage = 'design';
    wr4: SIZEOF(QUERY ( pdu <* QUERY ( pdr <* USEDIN(SELF,
             'METAL_CASTING.' +
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
              | ('METAL_CASTING.PRODUCT_DEFINITION_USAGE' IN TYPEOF(pdr)) )
              | (NOT ('METAL_CASTING.INVESTMENT_MOLD_DEFINITION' IN
             TYPEOF(pdu.relating_product_definition))) )) = 0;
    wr5: SIZEOF(QUERY ( acu <* QUERY ( pdr <* USEDIN(SELF,
             'METAL_CASTING.' +
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
              | ('METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr)) )
              | ('METAL_CASTING.INVESTMENT_MOLD_SPRUE_DEFINITION' IN
             TYPEOF(acu.related_product_definition)) )) = 1;
    wr6: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
              | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
             pdr.related_product_definition)) AND (pdr.
             related_product_definition.description =
             'investment pattern')) )) >= 1;
END_ENTITY; -- investment_mold_pattern_definition

ENTITY investment_mold_sprue_definition
  SUBTYPE OF (product_definition);
  WHERE
    wr1: SELF.frame_of_reference.name = 'investment casting';
    wr2: SELF.frame_of_reference.frame_of_reference.application =
             'casting';
    wr3: SELF.frame_of_reference.life_cycle_stage = 'design';
    wr4: SIZEOF(QUERY ( acu <* QUERY ( pdr <* USEDIN(SELF,
             'METAL_CASTING.' +
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
              | ('METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr)) )
              | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
             acu.related_product_definition)) AND (acu.description =
             'sprue component')) )) >= 1;
    wr5: SIZEOF(QUERY ( acu <* QUERY ( pdr <* USEDIN(SELF,
             'METAL_CASTING.' +
```

```
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
               | ('METAL_CASTING.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr)) )
               | ('METAL_CASTING.INVESTMENT_MOLD_PATTERN_DEFINITION' IN
              TYPEOF(acu.relating_product_definition)) )) >= 1;
END_ENTITY; -- investment_mold_sprue_definition

ENTITY length_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: 'METAL_CASTING.LENGTH_UNIT' IN TYPEOF(SELF\measure_with_unit.
              unit_component);
END_ENTITY; -- length_measure_with_unit

ENTITY length_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: (SELF\named_unit.dimensions.length_exponent = 1) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 0) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- length_unit

ENTITY line
  SUBTYPE OF (curve);
    pnt : cartesian_point;
    dir : vector;
  WHERE
    wr1: dir.dim = pnt.dim;
END_ENTITY; -- line

ENTITY linear_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: SELF\geometric_tolerance.name = 'linear profile';
    wr2: (NOT (('FEATURE_BASED_PROCESS_PLANNING.' +
              'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF)))
              OR (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
              datum_system) <= 3);
END_ENTITY; -- linear_profile_tolerance

ENTITY local_time;
    hour_component   : hour_in_day;
    minute_component : OPTIONAL minute_in_hour;
    second_component : OPTIONAL second_in_minute;
    zone             : coordinated_universal_time_offset;
  WHERE
    wr1: valid_time(SELF);
END_ENTITY; -- local_time
```

```
ENTITY loop
  SUPERTYPE OF (ONEOF (vertex_loop,edge_loop,poly_loop))
  SUBTYPE OF (topological_representation_item);
END_ENTITY; -- loop

ENTITY lot_effectivity
  SUBTYPE OF (effectivity);
    effectivity_lot_id   : identifier;
    effectivity_lot_size : measure_with_unit;
END_ENTITY; -- lot_effectivity

ENTITY make_from_usage_option
  SUBTYPE OF (product_definition_usage);
    ranking           : INTEGER;
    ranking_rationale : text;
    quantity          : measure_with_unit;
  WHERE
    wr1: ranking > 0;
END_ENTITY; -- make_from_usage_option

ENTITY manifold_solid_brep
  SUBTYPE OF (solid_model);
    outer : closed_shell;
END_ENTITY; -- manifold_solid_brep

ENTITY mapped_item
  SUBTYPE OF (representation_item);
    mapping_source : representation_map;
    mapping_target : representation_item;
  WHERE
    wr1: acyclic_mapped_representation(using_representations(SELF),[SELF]);
END_ENTITY; -- mapped_item

ENTITY material_property
  SUBTYPE OF (property_definition);
  UNIQUE
    ur1 : name, definition;
  WHERE
    wr1: ('METAL_CASTING.CHARACTERIZED_OBJECT' IN (TYPEOF(SELF\
             property_definition.definition) OR SIZEOF(bag_to_set(USEDIN(
             SELF,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) -
             QUERY ( temp <* bag_to_set(USEDIN(SELF,
             'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) | ((
             'MATERIAL_PROPERTY_REPRESENTATION_SCHEMA.' +
             'MATERIAL_PROPERTY_REPRESENTATION') IN TYPEOF(temp)) )))) =
             0;
END_ENTITY; -- material_property

ENTITY measure_qualification;
```

```
    name             : label;
    description      : text;
    qualified_measure : measure_with_unit;
    qualifiers       : SET [1:?] OF value_qualifier;
  WHERE
    wr1: SIZEOF(QUERY ( temp <* qualifiers | (
             'METAL_CASTING.PRECISION_QUALIFIER' IN TYPEOF(temp)) )) < 2;
END_ENTITY; -- measure_qualification

ENTITY measure_representation_item
  SUBTYPE OF (representation_item, measure_with_unit);
END_ENTITY; -- measure_representation_item

ENTITY measure_with_unit
  SUPERTYPE OF (ONEOF (length_measure_with_unit,
      plane_angle_measure_with_unit,ratio_measure_with_unit));
    value_component : measure_value;
    unit_component  : unit;
  WHERE
    wr1: valid_units(SELF);
END_ENTITY; -- measure_with_unit

ENTITY named_unit
  SUPERTYPE OF (ONEOF (length_unit,plane_angle_unit,ratio_unit));
    dimensions : dimensional_exponents;
END_ENTITY; -- named_unit

ENTITY offset_curve_2d
  SUBTYPE OF (curve);
    basis_curve    : curve;
    distance       : length_measure;
    self_intersect : LOGICAL;
  WHERE
    wr1: basis_curve.dim = 2;
END_ENTITY; -- offset_curve_2d

ENTITY open_shell
  SUBTYPE OF (connected_face_set);
END_ENTITY; -- open_shell

ENTITY organization;
    id          : OPTIONAL identifier;
    name        : label;
    description : text;
END_ENTITY; -- organization

ENTITY organization_assignment
  ABSTRACT SUPERTYPE;
    assigned_organization : organization;
    role                  : organization_role;
END_ENTITY; -- organization_assignment
```

```
ENTITY organization_role;
    name : label;
END_ENTITY; -- organization_role

ENTITY oriented_closed_shell
  SUBTYPE OF (closed_shell);
    closed_shell_element : closed_shell;
    orientation          : BOOLEAN;
  DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                        conditional_reverse(SELF.
                                        orientation,SELF.
                                        closed_shell_element.cfs_faces);
  WHERE
    wr1: NOT ('METAL_CASTING.ORIENTED_CLOSED_SHELL' IN TYPEOF(SELF.
             closed_shell_element));
END_ENTITY; -- oriented_closed_shell

ENTITY oriented_edge
  SUBTYPE OF (edge);
    edge_element : edge;
    orientation  : BOOLEAN;
  DERIVE
    SELF\edge.edge_start : vertex := boolean_choose(SELF.orientation,
                              SELF.edge_element.edge_start,SELF.
                              edge_element.edge_end);
    SELF\edge.edge_end   : vertex := boolean_choose(SELF.orientation,
                              SELF.edge_element.edge_end,SELF.
                              edge_element.edge_start);
  WHERE
    wr1: NOT ('METAL_CASTING.ORIENTED_EDGE' IN TYPEOF(SELF.edge_element));
END_ENTITY; -- oriented_edge

ENTITY oriented_face
  SUBTYPE OF (face);
    face_element : face;
    orientation  : BOOLEAN;
  DERIVE
    SELF\face.bounds : SET [1:?] OF face_bound := conditional_reverse(
                          SELF.orientation,SELF.face_element.bounds);
  WHERE
    wr1: NOT ('METAL_CASTING.ORIENTED_FACE' IN TYPEOF(SELF.face_element));
END_ENTITY; -- oriented_face

ENTITY oriented_open_shell
  SUBTYPE OF (open_shell);
    open_shell_element : open_shell;
    orientation        : BOOLEAN;
  DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                        conditional_reverse(SELF.
                                        orientation,SELF.
```

```
                                              open_shell_element.cfs_faces);
  WHERE
    wr1: NOT ('METAL_CASTING.ORIENTED_OPEN_SHELL' IN TYPEOF(SELF.
             open_shell_element));
END_ENTITY; -- oriented_open_shell

ENTITY oriented_path
  SUBTYPE OF (path);
    path_element : path;
    orientation  : BOOLEAN;
  DERIVE
    SELF\path.edge_list : LIST [1:?] OF UNIQUE oriented_edge :=
                          conditional_reverse(SELF.orientation,SELF.
                          path_element.edge_list);
  WHERE
    wr1: NOT ('METAL_CASTING.ORIENTED_PATH' IN TYPEOF(SELF.path_element));
END_ENTITY; -- oriented_path

ENTITY parabola
  SUBTYPE OF (conic);
    focal_dist : length_measure;
  WHERE
    wr1: focal_dist <> 0;
END_ENTITY; -- parabola

ENTITY parallelism_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SELF\geometric_tolerance.name = 'parallelism';
    wr2: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
             datum_system) = 1;
END_ENTITY; -- parallelism_tolerance

ENTITY parametric_representation_context
  SUBTYPE OF (representation_context);
END_ENTITY; -- parametric_representation_context

ENTITY path
  SUPERTYPE OF (ONEOF (edge_loop,oriented_path))
  SUBTYPE OF (topological_representation_item);
    edge_list : LIST [1:?] OF UNIQUE oriented_edge;
  WHERE
    wr1: path_head_to_tail(SELF);
END_ENTITY; -- path

ENTITY pcurve
  SUBTYPE OF (curve);
    basis_surface      : surface;
    reference_to_curve : definitional_representation;
  WHERE
    wr1: SIZEOF(reference_to_curve\representation.items) = 1;
    wr2: 'METAL_CASTING.CURVE' IN TYPEOF(reference_to_curve\
```

```
             representation.items[1]);
    wr3: reference_to_curve\representation.items[1]\
             geometric_representation_item.dim = 2;
END_ENTITY; -- pcurve

ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SELF\geometric_tolerance.name = 'perpendicularity';
    wr2: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
             datum_system) = 1;
END_ENTITY; -- perpendicularity_tolerance

ENTITY person;
    id            : identifier;
    last_name     : OPTIONAL label;
    first_name    : OPTIONAL label;
    middle_names  : OPTIONAL LIST [1:?] OF label;
    prefix_titles : OPTIONAL LIST [1:?] OF label;
    suffix_titles : OPTIONAL LIST [1:?] OF label;
  UNIQUE
    ur1 : id;
  WHERE
    wr1: EXISTS(last_name) OR EXISTS(first_name);
END_ENTITY; -- person

ENTITY person_and_organization;
    the_person       : person;
    the_organization : organization;
END_ENTITY; -- person_and_organization

ENTITY person_and_organization_assignment
  ABSTRACT SUPERTYPE;
    assigned_person_and_organization : person_and_organization;
    role                             : person_and_organization_role;
END_ENTITY; -- person_and_organization_assignment

ENTITY person_and_organization_role;
    name : label;
END_ENTITY; -- person_and_organization_role

ENTITY physical_structure_element
  SUBTYPE OF (product_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
              | ((
             'METAL_CASTING.PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP' IN
              TYPEOF(pdr)) AND (SIZEOF(QUERY ( prpc <* USEDIN(pdr.
             relating_product_definition.formation.of_product,
             'METAL_CASTING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')
              | (prpc.name = 'material') )) = 1)) )) >= 1;
```

```
    wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
              | (NOT (pdr.relating_product_definition.formation :=: pdr.
             related_product_definition.formation)) )) = 0;
END_ENTITY; -- physical_structure_element

ENTITY placement
  SUPERTYPE OF (ONEOF (axis1_placement,axis2_placement_2d,
      axis2_placement_3d))
  SUBTYPE OF (geometric_representation_item);
    location : cartesian_point;
END_ENTITY; -- placement

ENTITY plane
  SUBTYPE OF (elementary_surface);
END_ENTITY; -- plane

ENTITY plane_angle_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: 'METAL_CASTING.PLANE_ANGLE_UNIT' IN TYPEOF(SELF\
             measure_with_unit.unit_component);
END_ENTITY; -- plane_angle_measure_with_unit

ENTITY plane_angle_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: (SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
             named_unit.dimensions.mass_exponent = 0) AND (SELF\
             named_unit.dimensions.time_exponent = 0) AND (SELF\
             named_unit.dimensions.electric_current_exponent = 0) AND (
             SELF\named_unit.dimensions.
             thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
             .dimensions.amount_of_substance_exponent = 0) AND (SELF\
             named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- plane_angle_unit

ENTITY point
  SUPERTYPE OF (ONEOF (cartesian_point,point_on_curve))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- point

ENTITY point_on_curve
  SUBTYPE OF (point);
    basis_curve     : curve;
    point_parameter : parameter_value;
END_ENTITY; -- point_on_curve

ENTITY poly_loop
  SUBTYPE OF (loop, geometric_representation_item);
    polygon : LIST [3:?] OF UNIQUE cartesian_point;
END_ENTITY; -- poly_loop
```

```
ENTITY polyline
  SUBTYPE OF (bounded_curve);
    points : LIST [2:?] OF cartesian_point;
END_ENTITY; -- polyline

ENTITY position_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SELF\geometric_tolerance.name = 'position';
    wr2: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
             datum_system) <= 3;
END_ENTITY; -- position_tolerance

ENTITY process_plan_input_assignment
  SUBTYPE OF (action_assignment);
    items : SET [1:?] OF process_plan_input_item;
  WHERE
    wr1: 'METAL_CASTING.PROCESS_PLAN_VERSION' IN TYPEOF(SELF.
             assigned_action);
END_ENTITY; -- process_plan_input_assignment

ENTITY process_plan_version
  SUBTYPE OF (action);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,
             'METAL_CASTING.CASTING_DATE_ASSIGNMENT.ITEMS')) + SIZEOF(
             USEDIN(SELF,'METAL_CASTING.' +
             'CASTING_DATE_AND_TIME_ASSIGNMENT.ITEMS'))) = 1;
END_ENTITY; -- process_plan_version

ENTITY process_product_association;
    name            : label;
    description     : text;
    defined_product : characterized_product_definition;
    process         : product_definition_process;
END_ENTITY; -- process_product_association

ENTITY process_property_association;
    name              : label;
    description       : text;
    process           : property_process;
    property_or_shape : property_or_shape_select;
END_ENTITY; -- process_property_association

ENTITY product;
    id                 : identifier;
    name               : label;
    description        : text;
    frame_of_reference : SET [1:?] OF product_context;
  UNIQUE
    ur1 : id;
```

```
END_ENTITY; -- product

ENTITY product_category;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- product_category

ENTITY product_category_relationship;
    name         : label;
    description  : text;
    category     : product_category;
    sub_category : product_category;
  WHERE
    wr1: acyclic_product_category_relationship(SELF,[SELF.sub_category]);
END_ENTITY; -- product_category_relationship

ENTITY product_context
  SUBTYPE OF (application_context_element);
    discipline_type : label;
END_ENTITY; -- product_context

ENTITY product_definition;
    id                 : identifier;
    description        : text;
    formation          : product_definition_formation;
    frame_of_reference : product_definition_context;
END_ENTITY; -- product_definition

ENTITY product_definition_context
  SUBTYPE OF (application_context_element);
    life_cycle_stage : label;
END_ENTITY; -- product_definition_context

ENTITY product_definition_formation;
    id          : identifier;
    description : text;
    of_product  : product;
  UNIQUE
    ur1 : id, of_product;
END_ENTITY; -- product_definition_formation

ENTITY product_definition_process
  SUBTYPE OF (action);
    identification : identifier;
  INVERSE
    product_definitions : SET [1:?] OF process_product_association FOR
                          process;
END_ENTITY; -- product_definition_process

ENTITY product_definition_relationship;
    id                          : identifier;
    name                        : label;
```

```
    description                 : text;
    relating_product_definition : product_definition;
    related_product_definition  : product_definition;
END_ENTITY; -- product_definition_relationship

ENTITY product_definition_shape
  SUBTYPE OF (property_definition);
  UNIQUE
    ur1 : definition;
  WHERE
    wr1: 'METAL_CASTING.CHARACTERIZED_PRODUCT_DEFINITION' IN TYPEOF(SELF
             \property_definition.definition);
END_ENTITY; -- product_definition_shape

ENTITY product_definition_usage
  SUPERTYPE OF (ONEOF (make_from_usage_option,assembly_component_usage))
  SUBTYPE OF (product_definition_relationship);
  UNIQUE
    ur1 : id, relating_product_definition, related_product_definition;
  WHERE
    wr1: acyclic_product_definition_relationship(SELF,[SELF\
             product_definition_relationship.related_product_definition],
             'METAL_CASTING.PRODUCT_DEFINITION_USAGE.' +
             'RELATED_PRODUCT_DEFINITION');
END_ENTITY; -- product_definition_usage

ENTITY product_material_composition_relationship
  SUBTYPE OF (product_definition_relationship);
    class                : label;
    constituent_amount   : SET [1:?] OF measure_with_unit;
    composition_basis    : label;
    determination_method : text;
END_ENTITY; -- product_material_composition_relationship

ENTITY product_related_product_category
  SUBTYPE OF (product_category);
    products : SET [1:?] OF product;
END_ENTITY; -- product_related_product_category

ENTITY property_definition;
    name        : label;
    description : text;
    definition  : characterized_definition;
END_ENTITY; -- property_definition

ENTITY property_definition_relationship;
    name                         : label;
    description                  : text;
    relating_property_definition : property_definition;
    related_property_definition  : property_definition;
END_ENTITY; -- property_definition_relationship
```

```
ENTITY property_definition_representation;
    definition          : property_definition;
    used_representation : representation;
END_ENTITY; -- property_definition_representation

ENTITY property_inspection_or_test_requirement
  SUBTYPE OF (property_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
              'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
              | (pdr.name = 'inspection or test basis') )) = 1;
    wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_RELATIONSHIP.' +
             'RELATED_PROPERTY_DEFINITION') | (NOT (pdr.
             relating_property_definition.description IN [
             'composition requirement','shape requirement',
             'tolerance requirement','process requirement',
             'heat treat process requirement','property requirement',
             'required machining allowance',
             'surface roughness requirement'])) )) = 0;
    wr3: SIZEOF(USEDIN(SELF,'CASTING_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1;
    wr4: (NOT (SELF.name = 'sampling run size')) OR ((
             'METAL_CASTING.CASTING_PART_DEFINITION' IN TYPEOF(SELF.
             definition)) AND (SELF.definition.frame_of_reference.name IN
              ['customer part','foundary part']));
END_ENTITY; -- property_inspection_or_test_requirement

ENTITY property_process
  SUBTYPE OF (action);
    identification : identifier;
  INVERSE
    properties : SET [1:?] OF process_property_association FOR process;
END_ENTITY; -- property_process

ENTITY qualified_representation_item
  SUBTYPE OF (representation_item);
    qualifiers : SET [1:?] OF value_qualifier;
  WHERE
    wr1: SIZEOF(QUERY ( temp <* qualifiers | (
             'METAL_CASTING.PRECISION_QUALIFIER' IN TYPEOF(temp)) )) < 2;
END_ENTITY; -- qualified_representation_item

ENTITY quasi_uniform_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- quasi_uniform_curve

ENTITY quasi_uniform_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- quasi_uniform_surface

ENTITY ratio_measure_with_unit
```

```
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: 'METAL_CASTING.RATIO_UNIT' IN TYPEOF(SELF\measure_with_unit.
             unit_component);
END_ENTITY; -- ratio_measure_with_unit

ENTITY ratio_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: (SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
             named_unit.dimensions.mass_exponent = 0) AND (SELF\
             named_unit.dimensions.time_exponent = 0) AND (SELF\
             named_unit.dimensions.electric_current_exponent = 0) AND (
             SELF\named_unit.dimensions.
             thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
             .dimensions.amount_of_substance_exponent = 0) AND (SELF\
             named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- ratio_unit

ENTITY rational_b_spline_curve
  SUBTYPE OF (b_spline_curve);
    weights_data : LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:upper_index_on_control_points] OF REAL :=
                 list_to_array(weights_data,0,
                 upper_index_on_control_points);
  WHERE
    wr1: SIZEOF(weights_data) = SIZEOF(SELF\b_spline_curve.
             control_points_list);
    wr2: curve_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_curve

ENTITY rational_b_spline_surface
  SUBTYPE OF (b_spline_surface);
    weights_data : LIST [2:?] OF LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF REAL :=
                 make_array_of_array(weights_data,0,u_upper,0,v_upper);
  WHERE
    wr1: (SIZEOF(weights_data) = SIZEOF(SELF\b_spline_surface.
             control_points_list)) AND (SIZEOF(weights_data[1]) = SIZEOF(
             SELF\b_spline_surface.control_points_list[1]));
    wr2: surface_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_surface

ENTITY referenced_property_exception
  SUBTYPE OF (property_definition);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'METAL_CASTING.' +
             'CASTING_DOCUMENT_REFERENCE.ITEMS')) >= 1;
    wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
             'METAL_CASTING.PROPERTY_DEFINITION_RELATIONSHIP.' +
```

```
                'RELATED_PROPERTY_DEFINITION') | (NOT ((pdr.
                relating_property_definition.description IN [
                'composition requirement','shape requirement',
                'tolerance requirement','process requirement',
                'heat treat process requirement','property requirement',
                'required machining allowance',
                'surface roughness requirement']) OR (SIZEOF(TYPEOF(pdr.
                relating_property_definition) * [
                'METAL_CASTING.PROPERTY_INSPECTION_OR_TEST_REQUIREMENT',
                'METAL_CASTING.REPORTING_REQUIREMENT']) = 1))) )) = 0;
    wr3: SIZEOF(USEDIN(SELF,'CASTING_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1;
    wr4: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'CASTING_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
                SIZEOF(pdr.used_representation.items) = 1)) )) = 0;
    wr5: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'CASTING_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
                SIZEOF(QUERY ( it <* pdr.used_representation.items | (NOT (
                'METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))) ))
                = 0)) )) = 0;
END_ENTITY; -- referenced_property_exception

ENTITY reporting_requirement
  SUBTYPE OF (property_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
                 'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
                 | (pdr.name = 'reporting basis') )) = 1;
    wr2: SIZEOF(USEDIN(SELF,'CASTING_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1;
    wr3: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
                'PROPERTY_DEFINITION_RELATIONSHIP.' +
                'RELATED_PROPERTY_DEFINITION') | (NOT ((pdr.
                relating_property_definition.description IN [
                'composition requirement','shape requirement',
                'tolerance requirement','process requirement',
                'heat treat process requirement','property requirement',
                'required machining allowance',
                'surface roughness requirement']) OR (
                'METAL_CASTING.PROPERTY_INSPECTION_OR_TEST_REQUIREMENT' IN
                TYPEOF(pdr.relating_property_definition)))) )) = 0;
    wr4: ('METAL_CASTING.CASTING_PART_DEFINITION' IN TYPEOF(SELF.
                definition)) AND (SELF.definition.frame_of_reference.name IN
                 ['foundary part','customer part','finished']);
END_ENTITY; -- reporting_requirement

ENTITY representation;
    name            : label;
    items           : SET [1:?] OF representation_item;
    context_of_items : representation_context;
END_ENTITY; -- representation
```

```
ENTITY representation_context;
    context_identifier : identifier;
    context_type       : text;
  INVERSE
    representations_in_context : SET [1:?] OF representation FOR
                                    context_of_items;
END_ENTITY; -- representation_context

ENTITY representation_item;
    name : label;
  WHERE
    wr1: SIZEOF(using_representations(SELF)) > 0;
END_ENTITY; -- representation_item

ENTITY representation_map;
    mapping_origin        : representation_item;
    mapped_representation : representation;
  INVERSE
    map_usage : SET [1:?] OF mapped_item FOR mapping_source;
  WHERE
    wr1: item_in_context(SELF.mapping_origin,SELF.mapped_representation.
             context_of_items);
END_ENTITY; -- representation_map

ENTITY representation_relationship;
    name        : label;
    description : text;
    rep_1       : representation;
    rep_2       : representation;
END_ENTITY; -- representation_relationship

ENTITY resource_property;
    name        : label;
    description : text;
    resource    : characterized_resource_definition;
END_ENTITY; -- resource_property

ENTITY resource_property_representation;
    name           : label;
    description    : text;
    property       : resource_property;
    representation : representation;
END_ENTITY; -- resource_property_representation

ENTITY resource_requirement_type;
    name        : label;
    description : text;
END_ENTITY; -- resource_requirement_type

ENTITY sand_mold_definition
  SUBTYPE OF (product_definition);
  WHERE
```

```
    wr1: SELF.frame_of_reference.name = 'sand casting';
    wr2: SELF.frame_of_reference.frame_of_reference.application =
              'casting';
    wr3: SELF.frame_of_reference.life_cycle_stage = 'design';
    wr4: (1 <= SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
               'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
               | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
              pdr.related_product_definition)) AND (pdr.
              related_product_definition.description =
              'sand mold component')) ))) AND (SIZEOF(QUERY ( pdr <*
              USEDIN(SELF,'METAL_CASTING.' +
               'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
               | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
              pdr.related_product_definition)) AND (pdr.
              related_product_definition.description =
              'sand mold component')) )) <= 2);
    wr5: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
               'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
               | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
              pdr.related_product_definition)) AND (pdr.
              related_product_definition.description =
              'sand mold component') AND (NOT (pdr.name IN ['cope','drag']))) ))
              = 0;
    wr6: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
               'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
               | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
              pdr.related_product_definition)) AND (pdr.
              related_product_definition.description = 'flask')) )) = 1;
    wr7: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
               'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
               | (('METAL_CASTING.CASTING_MOLD_DESIGN_ELEMENT' IN TYPEOF(
              pdr.related_product_definition)) AND (pdr.
              related_product_definition.description =
              'pattern and rigging assembly')) )) = 1;
END_ENTITY; -- sand_mold_definition

ENTITY sequential_method
  SUBTYPE OF (serial_action_method);
    sequence_position : count_measure;
END_ENTITY; -- sequential_method

ENTITY serial_action_method
  SUBTYPE OF (action_method_relationship);
END_ENTITY; -- serial_action_method

ENTITY shape_aspect;
    name                  : label;
    description           : text;
    of_shape              : product_definition_shape;
    product_definitional  : LOGICAL;
END_ENTITY; -- shape_aspect
```

```
ENTITY shape_aspect_relationship;
    name                   : label;
    description            : text;
    relating_shape_aspect  : shape_aspect;
    related_shape_aspect   : shape_aspect;
END_ENTITY; -- shape_aspect_relationship

ENTITY shape_definition_representation
  SUBTYPE OF (property_definition_representation);
  WHERE
    wr1: ('METAL_CASTING.SHAPE_DEFINITION' IN TYPEOF(SELF.definition.
             definition)) OR ('METAL_CASTING.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(SELF.definition));
    wr2: 'METAL_CASTING.SHAPE_REPRESENTATION' IN TYPEOF(SELF.
             used_representation);
END_ENTITY; -- shape_definition_representation

ENTITY shape_representation
  SUBTYPE OF (representation);
END_ENTITY; -- shape_representation

ENTITY simulation_input
  SUBTYPE OF (property_definition);
  WHERE
    wr1: SIZEOF(QUERY ( caa <* USEDIN(SELF,
             'METAL_CASTING.CASTING_ACTION_ASSIGNMENT.ITEMS') | (
             'METAL_CASTING.SIMULATION_RUN' IN TYPEOF(caa.assigned_action)) ))
             >= 1;
    wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
              'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
              | ('METAL_CASTING.SIMULATION_INPUT_REGION' IN TYPEOF(pdr.
             related_property_definition)) )) >= 1;
    wr4: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
             used_representation.name = 'integration interval') )) >= 1;
END_ENTITY; -- simulation_input

ENTITY simulation_input_region
  SUBTYPE OF (property_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'METAL_CASTING.SHAPE_REPRESENTATION' IN TYPEOF(pdr.
             used_representation)) AND (pdr.used_representation.name =
             'region shape')) )) = 1;
    wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
             QUERY ( it <* pdr.used_representation.items | ((
             'METAL_CASTING.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
             AND (it.name = 'region type')) )) = 1) )) = 1;
    wr3: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
              'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
```

```
                 | ('METAL_CASTING.SIMULATION_INPUT' IN TYPEOF(pdr.
                relating_property_definition)) )) >= 1;
END_ENTITY; -- simulation_input_region

ENTITY simulation_output_region
  SUBTYPE OF (property_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
                 'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
                 | ('METAL_CASTING.SIMULATION_INPUT_REGION' IN TYPEOF(pdr.
                relating_property_definition)) )) = 1;
    wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'METAL_CASTING.SHAPE_REPRESENTATION' IN TYPEOF(pdr.
                used_representation)) AND (pdr.used_representation.name =
                'region shape')) )) = 1;
    wr3: SIZEOF(QUERY ( pdr <* USEDIN(SELF,'METAL_CASTING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
                'METAL_CASTING.SIMULATION_UNIT_STATE' IN TYPEOF(pdr.
                used_representation)) )) >= 1;
END_ENTITY; -- simulation_output_region

ENTITY simulation_run
  SUBTYPE OF (action);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'METAL_CASTING.ACTION_RESOURCE.USAGE')) = 1;
    wr2: SIZEOF(QUERY ( ar <* USEDIN(SELF,
                'METAL_CASTING.ACTION_RESOURCE.USAGE') | (NOT (ar.kind.name
                = 'simulation software')) )) = 0;
    wr3: SIZEOF(QUERY ( caa <* QUERY ( aa <* USEDIN(SELF,
                'METAL_CASTING.ACTION_ASSIGNMENT.ASSIGNED_ACTION') | (
                'METAL_CASTING.CASTING_ACTION_ASSIGNMENT' IN TYPEOF(aa)) )
                 | (SIZEOF(QUERY ( it <* caa.items | (
                'METAL_CASTING.SIMULATION_INPUT' IN TYPEOF(it)) )) = 1) )) =
                 1;
END_ENTITY; -- simulation_run

ENTITY simulation_unit_state
  SUBTYPE OF (representation);
  WHERE
    wr1: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
                'METAL_CASTING.MEASURE_REPRESENTATION_ITEM',
                'METAL_CASTING.TIME_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2)
                AND (it.name = 'evaluation time')) )) = 1;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (
                'METAL_CASTING.TENSOR_REPRESENTATION_ITEM' IN TYPEOF(it)) ))
                >= 1;
END_ENTITY; -- simulation_unit_state

ENTITY solid_model
  SUPERTYPE OF (manifold_solid_brep)
  SUBTYPE OF (geometric_representation_item);
```

```
END_ENTITY; -- solid_model

ENTITY spherical_surface
  SUBTYPE OF (elementary_surface);
    radius : positive_length_measure;
END_ENTITY; -- spherical_surface

ENTITY statistical_measure
  SUBTYPE OF (representation);
  WHERE
    wr1: SIZEOF(SELF.items) = 3;
    wr2: SIZEOF(QUERY ( it <* SELF.items | ((
             'METAL_CASTING.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it))
             AND (it.name = 'mean')) )) = 1;
    wr3: SIZEOF(QUERY ( it <* SELF.items | ((
             'METAL_CASTING.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it))
             AND (it.name = 'variance')) )) = 1;
    wr4: SIZEOF(QUERY ( it <* SELF.items | ((
             'METAL_CASTING.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it))
             AND ('METAL_CASTING.COUNT_MEASURE' IN TYPEOF(it\
             measure_with_unit.value_component)) AND (it.name =
             'number of measurements')) )) = 1;
END_ENTITY; -- statistical_measure

ENTITY surface
  SUPERTYPE OF (ONEOF (elementary_surface,swept_surface,bounded_surface))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- surface

ENTITY surface_curve
  SUBTYPE OF (curve);
    curve_3d              : curve;
    associated_geometry   : LIST [1:2] OF pcurve_or_surface;
    master_representation : preferred_surface_curve_representation;
  DERIVE
    basis_surface : SET [1:2] OF surface := get_basis_surface(SELF);
  WHERE
    wr1: curve_3d.dim = 3;
    wr2: ('METAL_CASTING.PCURVE' IN TYPEOF(associated_geometry[1])) OR (
             master_representation <> pcurve_s1);
    wr3: ('METAL_CASTING.PCURVE' IN TYPEOF(associated_geometry[2])) OR (
             master_representation <> pcurve_s2);
    wr4: NOT ('METAL_CASTING.PCURVE' IN TYPEOF(curve_3d));
END_ENTITY; -- surface_curve

ENTITY surface_of_linear_extrusion
  SUBTYPE OF (swept_surface);
    extrusion_axis : vector;
END_ENTITY; -- surface_of_linear_extrusion

ENTITY surface_of_revolution
  SUBTYPE OF (swept_surface);
```

```
      axis_position : axis1_placement;
    DERIVE
      axis_line : line := line(axis_position.location,vector(axis_position
                     .z,1));
  END_ENTITY; -- surface_of_revolution

  ENTITY surface_profile_tolerance
    SUBTYPE OF (geometric_tolerance);
    WHERE
      wr1: SELF\geometric_tolerance.name = 'surface profile';
      wr2: (NOT ('FEATURE_BASED_PROCESS_PLANNING.' +
               'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE')) OR (SIZEOF(SELF
               \geometric_tolerance_with_datum_reference.datum_system) <= 3);
  END_ENTITY; -- surface_profile_tolerance

  ENTITY swept_surface
    SUPERTYPE OF (ONEOF (surface_of_linear_extrusion,surface_of_revolution))
    SUBTYPE OF (surface);
      swept_curve : curve;
  END_ENTITY; -- swept_surface

  ENTITY symmetry_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
      wr1: SELF\geometric_tolerance.name = 'symmetry';
      wr2: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
               datum_system) <= 3;
  END_ENTITY; -- symmetry_tolerance

  ENTITY tensor_representation_item
    SUBTYPE OF (representation_item);
      tensor_value : tensor_type;
  END_ENTITY; -- tensor_representation_item

  ENTITY topological_representation_item
    SUPERTYPE OF (ONEOF (vertex,edge,face_bound,face,connected_face_set,
        loop ANDOR path))
    SUBTYPE OF (representation_item);
  END_ENTITY; -- topological_representation_item

  ENTITY toroidal_surface
    SUBTYPE OF (elementary_surface);
      major_radius : positive_length_measure;
      minor_radius : positive_length_measure;
  END_ENTITY; -- toroidal_surface

  ENTITY total_runout_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
      wr1: SELF\geometric_tolerance.name = 'total runout';
      wr2: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
               datum_system) <= 2;
```

```
END_ENTITY; -- total_runout_tolerance

ENTITY trimmed_curve
  SUBTYPE OF (bounded_curve);
    basis_curve           : curve;
    trim_1                : SET [1:2] OF trimming_select;
    trim_2                : SET [1:2] OF trimming_select;
    sense_agreement       : BOOLEAN;
    master_representation : trimming_preference;
  WHERE
    wr1: (HIINDEX(trim_1) = 1) XOR (TYPEOF(trim_1[1]) <> TYPEOF(trim_1[2]));
    wr2: (HIINDEX(trim_2) = 1) XOR (TYPEOF(trim_2[1]) <> TYPEOF(trim_2[2]));
END_ENTITY; -- trimmed_curve

ENTITY type_qualifier;
    name : label;
END_ENTITY; -- type_qualifier

ENTITY uniform_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- uniform_curve

ENTITY uniform_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- uniform_surface

ENTITY vector
  SUBTYPE OF (geometric_representation_item);
    orientation : direction;
    magnitude   : length_measure;
  WHERE
    wr1: magnitude >= 0;
END_ENTITY; -- vector

ENTITY versioned_action_request;
    id          : identifier;
    version     : label;
    purpose     : text;
    description : text;
END_ENTITY; -- versioned_action_request

ENTITY vertex
  SUBTYPE OF (topological_representation_item);
END_ENTITY; -- vertex

ENTITY vertex_loop
  SUBTYPE OF (loop);
    loop_vertex : vertex;
END_ENTITY; -- vertex_loop

ENTITY vertex_point
  SUBTYPE OF (vertex, geometric_representation_item);
```

```
    vertex_geometry : point;
END_ENTITY; -- vertex_point

RULE annotation_requires_shape_aspect FOR (representation, shape_aspect);

WHERE
  wr1: SIZEOF(QUERY ( rep <* representation | ((NOT (rep.name =
           'shape aspect annotation')) OR (SIZEOF(QUERY ( sa <*
           shape_aspect | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(rep,
           'METAL_CASTING.PROPERTY_DEFINITION_REPRESENTATION.' +
           'USED_REPRESENTATION') | (sa :=: pdr.definition.definition) ))
           = 1)) )) = 0)) )) = 0;

END_RULE; -- annotation_requires_shape_aspect

RULE application_context_requires_ap_definition FOR (application_context,
           application_protocol_definition);

WHERE
  wr1: SIZEOF(QUERY ( ac <* application_context | (NOT (SIZEOF(
           QUERY ( apd <* application_protocol_definition | ((ac :=: apd.
           application) AND (apd.
           application_interpreted_model_schema_name = 'metal_casting')) ))
           = 1)) )) = 0;

END_RULE; -- application_context_requires_ap_definition

RULE approval_requires_approval_date_time FOR (approval,
           approval_date_time);

WHERE
  wr1: SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( adt <*
           approval_date_time | (app :=: adt.dated_approval) )) = 1)) ))
           = 0;

END_RULE; -- approval_requires_approval_date_time

RULE approval_requires_approval_person_organization FOR (approval,
           approval_person_organization);

WHERE
  wr1: SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( apo <*
           approval_person_organization | (app :=: apo.
           authorized_approval) )) = 1)) )) = 0;

END_RULE; -- approval_requires_approval_person_organization

RULE compatible_dimension FOR (cartesian_point, direction,
           representation_context, geometric_representation_context);

WHERE
  wr1: SIZEOF(QUERY ( x <* cartesian_point | (SIZEOF(QUERY ( y <*
```

```
          geometric_representation_context | (item_in_context(x,y) AND (
          HIINDEX(x.coordinates) <> y.coordinate_space_dimension)) )) >
          0) )) = 0;
  wr2: SIZEOF(QUERY ( x <* direction | (SIZEOF(QUERY ( y <*
          geometric_representation_context | (item_in_context(x,y) AND (
          HIINDEX(x.direction_ratios) <> y.coordinate_space_dimension)) ))
          > 0) )) = 0;

END_RULE; -- compatible_dimension

RULE dependent_instantiable_approval FOR (approval);

WHERE
  wr1: SIZEOF(QUERY ( ast <* approval | (NOT (SIZEOF(USEDIN(ast,''')) >=
          1)) )) = 0;

END_RULE; -- dependent_instantiable_approval

RULE design_request_requires_date_or_date_and_time FOR (
          casting_date_assignment, casting_date_and_time_assignment,
          versioned_action_request);

WHERE
  wr1: SIZEOF(QUERY ( vr <* versioned_action_request | (NOT ((SIZEOF(
          QUERY ( cda <* casting_date_assignment | (vr IN cda.items) ))
          + SIZEOF(QUERY ( cdta <* casting_date_assignment | (vr IN cdta
          .items) ))) = 1)) )) = 0;

END_RULE; -- design_request_requires_date_or_date_and_time

RULE design_request_requires_person_and_organization FOR (
          casting_person_and_organization_assignment,
          versioned_action_request);

WHERE
  wr1: SIZEOF(QUERY ( vr <* versioned_action_request | (NOT (SIZEOF(
          QUERY ( cpoa <* casting_person_and_organization_assignment | (
          vr IN cpoa.items) )) = 1)) )) = 0;

END_RULE; -- design_request_requires_person_and_organization

RULE design_update_requires_date_or_date_and_time FOR (
          casting_date_assignment, casting_date_and_time_assignment,
          directed_action);

WHERE
  wr1: SIZEOF(QUERY ( da <* directed_action | (NOT ((SIZEOF(
          QUERY ( cda <* casting_date_assignment | ((NOT (da.name =
          'design update')) OR (da IN cda.items)) )) + SIZEOF(
          QUERY ( cdta <* casting_date_assignment | ((NOT (da.name =
          'design update')) OR (da IN cdta.items)) ))) = 1)) )) = 0;
```

```
END_RULE; -- design_update_requires_date_or_date_and_time

RULE design_update_requires_item_definition FOR (
          casting_action_assignment);

WHERE
  wr1: SIZEOF(QUERY ( caa <* casting_action_assignment | ((caa.
          assigned_action.name = 'design update') AND (SIZEOF(caa.items)
          = 1) AND (SIZEOF(['METAL_CASTING.CHARACTERIZED_OBJECT',
          'METAL_CASTING.PRODUCT_DEFINITION',
          'METAL_CASTING.PROPERTY_DEFINITION',
          'METAL_CASTING.SHAPE_ASPECT'] * TYPEOF(caa.items)) = 1)) )) =
          1;

END_RULE; -- design_update_requires_item_definition

RULE design_update_requires_person_and_organization FOR (
          casting_person_and_organization_assignment, directed_action);

WHERE
  wr1: SIZEOF(QUERY ( da <* directed_action | (NOT (SIZEOF(
          QUERY ( cpoa <* casting_person_and_organization_assignment | (
          (NOT (da.name = 'design update')) OR (da IN cpoa.items)) )) =
          1)) )) = 0;

END_RULE; -- design_update_requires_person_and_organization

RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);

WHERE
  wr1: SIZEOF(QUERY ( gt <* geometric_tolerance | (SIZEOF(TYPEOF(gt) * [
          'METAL_CASTING.ANGULARITY_TOLERANCE',
          'METAL_CASTING.CIRCULAR_RUNOUT_TOLERANCE',
          'METAL_CASTING.CONCENTRICITY_TOLERANCE',
          'METAL_CASTING.LINEAR_PROFILE_TOLERANCE',
          'METAL_CASTING.PARALLELISM_TOLERANCE',
          'METAL_CASTING.PERPENDICULARITY_TOLERANCE',
          'METAL_CASTING.POSITIONAL_TOLERANCE',
          'METAL_CASTING.SURFACE_PROFILE_TOLERANCE',
          'METAL_CASTING.SYMMETRY_TOLERANCE',
          'METAL_CASTING.TOTAL_RUNOUT_TOLERANCE']) >= 2) )) = 0;

END_RULE; -- geometric_tolerance_subtype_exclusiveness

RULE item_requirement_requires_specification FOR (property_definition,
          casting_document_reference);

WHERE
  wr1: SIZEOF(QUERY ( pd <* property_definition | (NOT (SIZEOF(
          QUERY ( cdr <* casting_document_reference | (NOT ((pd.
          description IN ['composition requirement','shape requirement',
          'tolerance requirement','process requirement',
```

```
           'heat treat process requirement','property requirement',
           'required machining allowance','surface roughness requirement'])
           OR (SIZEOF(TYPEOF(pd) * [
           'METAL_CASTING.PROPERTY_INSPECTION_OR_TEST_REQUIREMENT',
           'METAL_CASTING.REPORTING_REQUIREMENT']) = 1) OR (pd IN cdr.
           items))) )) >= 1)) )) = 0;

END_RULE; -- item_requirement_requires_specification

RULE lot_requires_date_or_date_and_time FOR (casting_date_assignment,
           casting_date_and_time_assignment, lot_effectivity);

WHERE
  wr1: SIZEOF(QUERY ( le <* lot_effectivity | (NOT ((SIZEOF(
           QUERY ( cda <* casting_date_assignment | (le IN cda.items) ))
           + SIZEOF(QUERY ( cdta <* casting_date_assignment | (le IN cdta
           .items) ))) = 1)) )) = 0;

END_RULE; -- lot_requires_date_or_date_and_time

RULE process_execution_record_requires_process FOR (representation,
           action_method);

WHERE
  wr1: SIZEOF(QUERY ( rep <* representation | ((NOT (rep.name IN [
           'machine setting record','substance usage record',
           'substance composition element record',
           'process parameter record'])) OR (SIZEOF(QUERY ( am <*
           action_method | (NOT (SIZEOF(QUERY ( apr <* USEDIN(rep,
           'METAL_CASTING.ACTION_PROPERTY_REPRESENTATION.REPRESENTATION')
            | (am :=: apr.property) )) = 1)) )) = 0)) )) = 0;

END_RULE; -- process_execution_record_requires_process

RULE process_execution_requires_process_plan FOR (
           product_definition_process, property_process,
           action_relationship);

WHERE
  wr1: SIZEOF(QUERY ( pdp <* product_definition_process | (NOT (SIZEOF(
           QUERY ( ar <* action_relationship | ((
           'METAL_CASTING.EXECUTED_ACTION' IN TYPEOF(pdp)) AND (pdp :=:
           ar.related_action)) )) = 1)) )) = 0;
  wr2: SIZEOF(QUERY ( ppp <* property_process | (NOT (SIZEOF(
           QUERY ( ar <* action_relationship | ((
           'METAL_CASTING.EXECUTED_ACTION' IN TYPEOF(ppp)) AND (ppp :=:
           ar.related_action)) )) = 1)) )) = 0;

END_RULE; -- process_execution_requires_process_plan

RULE specification_requires_organization FOR (
           casting_organization_assignment, document);
```

```
WHERE
  wr1: SIZEOF(QUERY ( doc <* document | (NOT (SIZEOF(QUERY ( coa <*
           casting_organization_assignment | ((NOT (doc.kind.
           product_data_type = 'specification')) OR (doc IN coa.items)) ))
           = 1)) )) = 0;

END_RULE; -- specification_requires_organization

FUNCTION acyclic_mapped_representation(
             parent_set: SET OF representation;
             children_set: SET OF representation_item
    ): BOOLEAN;

  LOCAL
    i : INTEGER;
    x : SET OF representation_item;
    y : SET OF representation_item;
  END_LOCAL;
  x := QUERY ( z <* children_set | ('METAL_CASTING.MAPPED_ITEM' IN
      TYPEOF(z)) );
  IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
      IF x[i]\mapped_item.mapping_source.mapped_representation IN
          parent_set THEN
        RETURN(FALSE);
      END_IF;
      IF NOT acyclic_mapped_representation(parent_set + x[i]\mapped_item
          .mapping_source.mapped_representation,x[i]\mapped_item.
          mapping_source.mapped_representation.items) THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_IF;
  x := children_set - x;
  IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
      y := QUERY ( z <* bag_to_set(USEDIN(x[i],'')) | (
          'METAL_CASTING.REPRESENTATION_ITEM' IN TYPEOF(z)) );
      IF NOT acyclic_mapped_representation(parent_set,y) THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(TRUE);

END_FUNCTION; -- acyclic_mapped_representation

FUNCTION acyclic_product_category_relationship(
             relation: product_category_relationship;
             children: SET OF product_category
    ): LOGICAL;
```

```
  LOCAL
    i              : INTEGER;
    x              : SET OF product_category_relationship;
    local_children : SET OF product_category;
  END_LOCAL;
  REPEAT i := 1 TO HIINDEX(children) BY 1;
    IF relation.category :=: children[i] THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
  x := bag_to_set(USEDIN(relation.category,'METAL_CASTING.' +
      'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY'));
  local_children := children + relation.category;
  IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
      IF NOT acyclic_product_category_relationship(x[i],local_children)
           THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(TRUE);

END_FUNCTION; -- acyclic_product_category_relationship

FUNCTION acyclic_product_definition_relationship(
             relation: product_definition_relationship;
             relatives: SET OF product_definition;
             specific_relation: STRING
    ): LOGICAL;

  LOCAL
    i               : INTEGER;
    x               : SET OF product_definition_relationship;
    local_relatives : SET OF product_definition;
  END_LOCAL;
  REPEAT i := 1 TO HIINDEX(relatives) BY 1;
    IF relation.relating_product_definition :=: relatives[i] THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
  x := bag_to_set(USEDIN(relation.relating_product_definition,
      specific_relation));
  local_relatives := relatives + relation.relating_product_definition;
  IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
      IF NOT acyclic_product_definition_relationship(x[i],
          local_relatives,specific_relation) THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
```

```
  END_IF;
  RETURN(TRUE);

END_FUNCTION; -- acyclic_product_definition_relationship

FUNCTION associated_surface(
             arg: pcurve_or_surface
    ): surface;

  LOCAL
    surf : surface;
  END_LOCAL;
  IF 'METAL_CASTING.PCURVE' IN TYPEOF(arg) THEN
    surf := arg.basis_surface;
  ELSE
    surf := arg;
  END_IF;
  RETURN(surf);

END_FUNCTION; -- associated_surface

FUNCTION bag_to_set(
             the_bag: BAG OF GENERIC:intype
    ): SET OF GENERIC:intype;

  LOCAL
    i       : INTEGER;
    the_set : SET OF GENERIC:intype := [];
  END_LOCAL;
  IF SIZEOF(the_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(the_bag) BY 1;
      the_set := the_set + the_bag[i];
    END_REPEAT;
  END_IF;
  RETURN(the_set);

END_FUNCTION; -- bag_to_set

FUNCTION base_axis(
             dim: INTEGER;
             axis1, axis2, axis3: direction
    ): LIST [2:3] OF direction;

  LOCAL
    u      : LIST [2:3] OF direction;
    vec    : direction;
    factor : REAL;
  END_LOCAL;
  IF dim = 3 THEN
    u[3] := NVL(normalise(axis3),direction([0,0,1]));
    u[1] := first_proj_axis(u[3],axis1);
    u[2] := second_proj_axis(u[3],u[1],axis2);
```

```
  ELSE
    u[3] := ?;
    IF EXISTS(axis1) THEN
      u[1] := normalise(axis1);
      u[2] := orthogonal_complement(u[1]);
      IF EXISTS(axis2) THEN
        factor := dot_product(axis2,u[2]);
        IF factor < 0 THEN
          u[2].direction_ratios[1] := -u[2].direction_ratios[1];
          u[2].direction_ratios[2] := -u[2].direction_ratios[2];
        END_IF;
      END_IF;
    ELSE
      IF EXISTS(axis2) THEN
        u[2] := normalise(axis2);
        u[1] := orthogonal_complement(u[2]);
        u[1].direction_ratios[1] := -u[1].direction_ratios[1];
        u[1].direction_ratios[2] := -u[1].direction_ratios[2];
      ELSE
        u[1].direction_ratios[1] := 1;
        u[1].direction_ratios[2] := 0;
        u[2].direction_ratios[1] := 0;
        u[2].direction_ratios[2] := 1;
      END_IF;
    END_IF;
  END_IF;
  RETURN(u);

END_FUNCTION; -- base_axis

FUNCTION boolean_choose(
             b: BOOLEAN;
             choice1, choice2: GENERIC:item
    ): GENERIC:item;
  IF b THEN
    RETURN(choice1);
  ELSE
    RETURN(choice2);
  END_IF;

END_FUNCTION; -- boolean_choose

FUNCTION build_2axes(
             ref_direction: direction
    ): LIST [2:2] OF direction;

  LOCAL
    u : LIST [2:2] OF direction;
  END_LOCAL;
  u[1] := NVL(normalise(ref_direction),direction([1,0]));
  u[2] := orthogonal_complement(u[1]);
  RETURN(u);
```

```
END_FUNCTION; -- build_2axes

FUNCTION build_axes(
             axis, ref_direction: direction
    ): LIST [3:3] OF direction;

  LOCAL
    u : LIST [3:3] OF direction;
  END_LOCAL;
  u[3] := NVL(normalise(axis),direction([0,0,1]));
  u[1] := first_proj_axis(u[3],ref_direction);
  u[2] := normalise(cross_product(u[3],u[1])).orientation;
  RETURN(u);

END_FUNCTION; -- build_axes

FUNCTION conditional_reverse(
             p: BOOLEAN;
             an_item: reversible_topology
    ): reversible_topology;
  IF p THEN
    RETURN(an_item);
  ELSE
    RETURN(topology_reversed(an_item));
  END_IF;

END_FUNCTION; -- conditional_reverse

FUNCTION constraints_composite_curve_on_surface(
             c: composite_curve_on_surface
    ): BOOLEAN;

  LOCAL
    n_segments : INTEGER := SIZEOF(c.segments);
  END_LOCAL;
  REPEAT k := 1 TO n_segments BY 1;
    IF (NOT ('METAL_CASTING.PCURVE' IN TYPEOF(c\composite_curve.segments
        [k].parent_curve))) AND (NOT ('METAL_CASTING.SURFACE_CURVE' IN
        TYPEOF(c\composite_curve.segments[k].parent_curve))) AND (NOT (
        'METAL_CASTING.COMPOSITE_CURVE_ON_SURFACE' IN TYPEOF(c\
        composite_curve.segments[k].parent_curve))) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
  RETURN(TRUE);

END_FUNCTION; -- constraints_composite_curve_on_surface

FUNCTION constraints_param_b_spline(
             degree, up_knots, up_cp: INTEGER;
             knot_mult: LIST OF INTEGER;
```

```
            knots: LIST OF parameter_value
    ): BOOLEAN;

  LOCAL
    k      : INTEGER;
    l      : INTEGER;
    sum    : INTEGER;
    result : BOOLEAN := TRUE;
  END_LOCAL;
  sum := knot_mult[1];
  REPEAT i := 2 TO up_knots BY 1;
    sum := sum + knot_mult[i];
  END_REPEAT;
  IF (degree < 1) OR (up_knots < 2) OR (up_cp < degree) OR (sum <> (
      degree + up_cp + 2)) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
  k := knot_mult[1];
  IF (k < 1) OR (k > (degree + 1)) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
  REPEAT i := 2 TO up_knots BY 1;
    IF (knot_mult[i] < 1) OR (knots[i] <= knots[i - 1]) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
    k := knot_mult[i];
    IF (i < up_knots) AND (k > degree) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
    IF (i = up_knots) AND (k > (degree + 1)) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
  END_REPEAT;
  RETURN(result);

END_FUNCTION; -- constraints_param_b_spline

FUNCTION cross_product(
            arg1, arg2: direction
    ): vector;

  LOCAL
    v2     : LIST [3:3] OF REAL;
    v1     : LIST [3:3] OF REAL;
    mag    : REAL;
    res    : direction;
    result : vector;
```

```
  END_LOCAL;
  IF (NOT EXISTS(arg1)) OR (arg1.dim = 2) OR (NOT EXISTS(arg2)) OR (arg2
      .dim = 2) THEN
    RETURN(?);
  ELSE
    BEGIN
      v1 := normalise(arg1).direction_ratios;
      v2 := normalise(arg2).direction_ratios;
      res.direction_ratios[1] := (v1[2] * v2[3]) - (v1[3] * v2[2]);
      res.direction_ratios[2] := (v1[3] * v2[1]) - (v1[1] * v2[3]);
      res.direction_ratios[3] := (v1[1] * v2[2]) - (v1[2] * v2[1]);
      mag := 0;
      REPEAT i := 1 TO 3 BY 1;
        mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
      END_REPEAT;
      IF mag > 0 THEN
        result.orientation := res;
        result.magnitude := SQRT(mag);
      ELSE
        result.orientation := arg1;
        result.magnitude := 0;
      END_IF;
      RETURN(result);
    END;
  END_IF;

END_FUNCTION; -- cross_product

FUNCTION curve_weights_positive(
             b: rational_b_spline_curve
    ): BOOLEAN;

  LOCAL
    result : BOOLEAN := TRUE;
  END_LOCAL;
  REPEAT i := 0 TO b.upper_index_on_control_points BY 1;
    IF b.weights[i] <= 0 THEN
      result := FALSE;
      RETURN(result);
    END_IF;
  END_REPEAT;
  RETURN(result);

END_FUNCTION; -- curve_weights_positive

FUNCTION derive_dimensional_exponents(
             x: unit
    ): dimensional_exponents;

  LOCAL
    i      : INTEGER;
    result : dimensional_exponents := dimensional_exponents(0,0,0,0,0,0,
```

```
             0);
  END_LOCAL;
  IF 'METAL_CASTING.DERIVED_UNIT' IN TYPEOF(x) THEN
    REPEAT i := LOINDEX(x.elements) TO HIINDEX(x.elements) BY 1;
      result.length_exponent := result.length_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.length_exponent);
      result.mass_exponent := result.mass_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.mass_exponent);
      result.time_exponent := result.time_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.time_exponent);
      result.electric_current_exponent := result.
          electric_current_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.electric_current_exponent);
      result.thermodynamic_temperature_exponent := result.
          thermodynamic_temperature_exponent + (x.elements[i].exponent *
          x.elements[i].unit.dimensions.
          thermodynamic_temperature_exponent);
      result.amount_of_substance_exponent := result.
          amount_of_substance_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.amount_of_substance_exponent);
      result.luminous_intensity_exponent := result.
          luminous_intensity_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.luminous_intensity_exponent);
    END_REPEAT;
  ELSE
    result := x.dimensions;
  END_IF;
  RETURN(result);

END_FUNCTION; -- derive_dimensional_exponents

FUNCTION dimension_of(
             item: geometric_representation_item
    ): dimension_count;

  LOCAL
    x : SET OF representation;
    y : representation_context;
  END_LOCAL;
  x := using_representations(item);
  y := x[1].context_of_items;
  RETURN(y\geometric_representation_context.coordinate_space_dimension);

END_FUNCTION; -- dimension_of

FUNCTION dot_product(
             arg1, arg2: direction
    ): REAL;

  LOCAL
    ndim   : INTEGER;
    scalar : REAL;
```

```
    vec1   : direction;
    vec2   : direction;
  END_LOCAL;
  IF (NOT EXISTS(arg1)) OR (NOT EXISTS(arg2)) THEN
    scalar := ?;
  ELSE
    IF arg1.dim <> arg2.dim THEN
      scalar := ?;
    ELSE
      BEGIN
        vec1 := normalise(arg1);
        vec2 := normalise(arg2);
        ndim := arg1.dim;
        scalar := 0;
        REPEAT i := 1 TO ndim BY 1;
          scalar := scalar + (vec1.direction_ratios[i] * vec2.
              direction_ratios[i]);
        END_REPEAT;
      END;
    END_IF;
  END_IF;
  RETURN(scalar);

END_FUNCTION; -- dot_product

FUNCTION edge_reversed(
            an_edge: edge
    ): edge;

  LOCAL
    the_reverse : edge;
  END_LOCAL;
  IF 'METAL_CASTING.ORIENTED_EDGE' IN TYPEOF(an_edge) THEN
    the_reverse := oriented_edge(an_edge\oriented_edge.edge_element,NOT
        an_edge\oriented_edge.orientation);
  ELSE
    the_reverse := oriented_edge(an_edge,FALSE);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- edge_reversed

FUNCTION face_bound_reversed(
            a_face_bound: face_bound
    ): face_bound;

  LOCAL
    the_reverse : face_bound;
  END_LOCAL;
  IF 'METAL_CASTING.FACE_OUTER_BOUND' IN TYPEOF(a_face_bound) THEN
    the_reverse := face_bound(a_face_bound\face_bound.bound,NOT
        a_face_bound\face_bound.orientation);
```

```
  ELSE
    the_reverse := face_bound(a_face_bound.bound,NOT a_face_bound.
        orientation);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- face_bound_reversed

FUNCTION face_reversed(
             a_face: face
    ): face;

  LOCAL
    the_reverse : face;
  END_LOCAL;
  IF 'METAL_CASTING.ORIENTED_FACE' IN TYPEOF(a_face) THEN
    the_reverse := oriented_face(a_face\oriented_face.face_element,NOT
        a_face\oriented_face.orientation);
  ELSE
    the_reverse := oriented_face(a_face,FALSE);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- face_reversed

FUNCTION first_proj_axis(
             z_axis, arg: direction
    ): direction;

  LOCAL
    x_vec  : vector;
    v      : direction;
    z      : direction;
    x_axis : direction;
  END_LOCAL;
  IF (NOT EXISTS(z_axis)) OR (NOT EXISTS(arg)) OR (arg.dim <> 3) THEN
    x_axis := ?;
  ELSE
    z_axis := normalise(z_axis);
    IF NOT EXISTS(arg) THEN
      IF z_axis <> direction([1,0,0]) THEN
        v := direction([1,0,0]);
      ELSE
        v := direction([0,1,0]);
      END_IF;
    ELSE
      IF cross_product(arg,z).magnitude = 0 THEN
        RETURN(?);
      ELSE
        v := normalise(arg);
      END_IF;
    END_IF;
```

```
    x_vec := scalar_times_vector(dot_product(v,z),z_axis);
    x_axis := vector_difference(v,x_vec).orientation;
    x_axis := normalise(x_axis);
  END_IF;
  RETURN(x_axis);

END_FUNCTION; -- first_proj_axis

FUNCTION get_basis_surface(
             c: curve_on_surface
    ): SET [0:2] OF surface;

  LOCAL
    surfs : SET [0:2] OF surface;
    n     : INTEGER;
  END_LOCAL;
  surfs := [];
  IF 'METAL_CASTING.PCURVE' IN TYPEOF(c) THEN
    surfs := [c\pcurve.basis_surface];
  ELSE
    IF 'METAL_CASTING.SURFACE_CURVE' IN TYPEOF(c) THEN
      n := SIZEOF(c\surface_curve.associated_geometry);
      REPEAT i := 1 TO n BY 1;
        surfs := surfs + associated_surface(c\surface_curve.
            associated_geometry[i]);
      END_REPEAT;
    END_IF;
  END_IF;
  IF 'METAL_CASTING.COMPOSITE_CURVE_ON_SURFACE' IN TYPEOF(c) THEN
    n := SIZEOF(c\composite_curve_on_surface.segments);
    surfs := get_basis_surface(c\composite_curve_on_surface.segments[1].
        parent_curve);
    IF n > 1 THEN
      REPEAT i := 2 TO n BY 1;
        surfs := surfs * get_basis_surface(c\composite_curve_on_surface.
            segments[i].parent_curve);
      END_REPEAT;
    END_IF;
  END_IF;
  RETURN(surfs);

END_FUNCTION; -- get_basis_surface

FUNCTION item_in_context(
             item: representation_item;
             cntxt: representation_context
    ): BOOLEAN;

  LOCAL
    i : INTEGER;
    y : BAG OF representation_item;
  END_LOCAL;
```

```
  IF SIZEOF(USEDIN(item,'METAL_CASTING.REPRESENTATION.ITEMS') * cntxt.
      representations_in_context) > 0 THEN
    RETURN(TRUE);
  ELSE
    y := QUERY ( z <* USEDIN(item,'') | (
        'METAL_CASTING.REPRESENTATION_ITEM' IN TYPEOF(z)) );
    IF SIZEOF(y) > 0 THEN
      REPEAT i := 1 TO HIINDEX(y) BY 1;
        IF item_in_context(y[i],cntxt) THEN
          RETURN(TRUE);
        END_IF;
      END_REPEAT;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- item_in_context

FUNCTION leap_year(
             year: year_number
    ): BOOLEAN;
  IF (((year MOD 4) = 0) AND ((year MOD 100) <> 0)) OR ((year MOD 400) =
       0) THEN
    RETURN(TRUE);
  ELSE
    RETURN(FALSE);
  END_IF;

END_FUNCTION; -- leap_year

FUNCTION list_face_loops(
             f: face
    ): LIST [0:?] OF loop;

  LOCAL
    loops : LIST [0:?] OF loop := [];
  END_LOCAL;
  REPEAT i := 1 TO SIZEOF(f.bounds) BY 1;
    loops := loops + f.bounds[i].bound;
  END_REPEAT;
  RETURN(loops);

END_FUNCTION; -- list_face_loops

FUNCTION list_of_topology_reversed(
             a_list: list_of_reversible_topology_item
    ): list_of_reversible_topology_item;

  LOCAL
    the_reverse : list_of_reversible_topology_item;
  END_LOCAL;
  the_reverse := [];
```

```
    REPEAT i := 1 TO SIZEOF(a_list) BY 1;
      the_reverse := topology_reversed(a_list[i]) + the_reverse;
    END_REPEAT;
    RETURN(the_reverse);

  END_FUNCTION; -- list_of_topology_reversed

  FUNCTION list_to_array(
               lis: LIST [0:?] OF GENERIC:t;
               low, u: INTEGER
      ): ARRAY [low:u] OF GENERIC:t;

    LOCAL
      n   : INTEGER;
      res : ARRAY [low:u] OF GENERIC:t;
    END_LOCAL;
    n := SIZEOF(lis);
    IF n <> ((u - low) + 1) THEN
      RETURN(?);
    ELSE
      REPEAT i := 1 TO n BY 1;
        res[(low + i) - 1] := lis[i];
      END_REPEAT;
      RETURN(res);
    END_IF;

  END_FUNCTION; -- list_to_array

  FUNCTION list_to_set(
               l: LIST [0:?] OF GENERIC:t
      ): SET OF GENERIC:t;

    LOCAL
      s : SET OF GENERIC:t := [];
    END_LOCAL;
    REPEAT i := 1 TO SIZEOF(l) BY 1;
      s := s + l[i];
    END_REPEAT;
    RETURN(s);

  END_FUNCTION; -- list_to_set

  FUNCTION make_array_of_array(
               lis: LIST [1:?] OF LIST [1:?] OF GENERIC:t;
               low1, u1, low2, u2: INTEGER
      ): ARRAY [low1:u1] OF ARRAY [low2:u2] OF GENERIC:t;

    LOCAL
      n2   : INTEGER;
      n1   : INTEGER;
      res  : ARRAY [low1:u1] OF ARRAY [low2:u2] OF GENERIC:t;
      resl : LIST [1:?] OF ARRAY [low2:u2] OF GENERIC:t;
```

```
  END_LOCAL;
  n1 := SIZEOF(lis);
  n2 := SIZEOF(lis[1]);
  IF (n1 <> ((u1 - low1) + 1)) AND (n2 <> ((u2 - low2) + 1)) THEN
    RETURN(?);
  END_IF;
  REPEAT i := 1 TO n1 BY 1;
    IF SIZEOF(lis[i]) <> n2 THEN
      RETURN(?);
    END_IF;
  END_REPEAT;
  REPEAT i := 1 TO n1 BY 1;
    resl[i] := list_to_array(lis[i],low2,u2);
  END_REPEAT;
  res := list_to_array(resl,low1,u1);
  RETURN(res);

END_FUNCTION; -- make_array_of_array

FUNCTION mixed_loop_type_set(
             l: SET [0:?] OF loop
    ): LOGICAL;

  LOCAL
    i              : INTEGER;
    poly_loop_type : LOGICAL;
  END_LOCAL;
  IF SIZEOF(l) <= 1 THEN
    RETURN(FALSE);
  END_IF;
  poly_loop_type := 'METAL_CASTING.POLY_LOOP' IN TYPEOF(l[1]);
  REPEAT i := 2 TO SIZEOF(l) BY 1;
    IF ('METAL_CASTING.POLY_LOOP' IN TYPEOF(l[i])) <> poly_loop_type
         THEN
      RETURN(TRUE);
    END_IF;
  END_REPEAT;
  RETURN(FALSE);

END_FUNCTION; -- mixed_loop_type_set

FUNCTION msb_shells(
             brep: manifold_solid_brep;
             schema_name: STRING
    ): SET [1:?] OF closed_shell;
  IF (schema_name + '.BREP_WITH_VOIDS') IN TYPEOF(brep) THEN
    RETURN(brep\brep_with_voids.voids + brep.outer);
  ELSE
    RETURN([brep.outer]);
  END_IF;

END_FUNCTION; -- msb_shells
```

```
FUNCTION normalise(
             arg: vector_or_direction
    ): vector_or_direction;

  LOCAL
    ndim   : INTEGER;
    v      : direction;
    vec    : vector;
    mag    : REAL;
    result : vector_or_direction;
  END_LOCAL;
  IF NOT EXISTS(arg) THEN
    result := ?;
  ELSE
    ndim := arg.dim;
    IF 'METAL_CASTING.VECTOR' IN TYPEOF(arg) THEN
      BEGIN
        vec := arg;
        v := arg.orientation;
        IF arg.magnitude = 0 THEN
          RETURN(?);
        ELSE
          vec.magnitude := 1;
        END_IF;
      END;
    ELSE
      v := arg;
    END_IF;
    mag := 0;
    REPEAT i := 1 TO ndim BY 1;
      mag := mag + (v.direction_ratios[i] * v.direction_ratios[i]);
    END_REPEAT;
    IF mag > 0 THEN
      mag := SQRT(mag);
      REPEAT i := 1 TO ndim BY 1;
        v.direction_ratios[i] := v.direction_ratios[i] / mag;
      END_REPEAT;
      IF 'METAL_CASTING.VECTOR' IN TYPEOF(arg) THEN
        vec.orientation := v;
        result := vec;
      ELSE
        result := v;
      END_IF;
    ELSE
      RETURN(?);
    END_IF;
  END_IF;
  RETURN(result);

END_FUNCTION; -- normalise
```

```
FUNCTION orthogonal_complement(
             vec: direction
    ): direction;

  LOCAL
    result : direction;
  END_LOCAL;
  IF (vec.dim <> 2) OR (NOT EXISTS(vec)) THEN
    RETURN(?);
  ELSE
    result.direction_ratios[1] := -vec.direction_ratios[2];
    result.direction_ratios[2] := vec.direction_ratios[1];
    RETURN(result);
  END_IF;

END_FUNCTION; -- orthogonal_complement

FUNCTION path_head_to_tail(
             a_path: path
    ): LOGICAL;

  LOCAL
    n : INTEGER;
    p : BOOLEAN := TRUE;
  END_LOCAL;
  n := SIZEOF(a_path.edge_list);
  REPEAT i := 2 TO n BY 1;
    p := p AND (a_path.edge_list[i - 1].edge_end :=: a_path.edge_list[i]
        .edge_start);
  END_REPEAT;
  RETURN(p);

END_FUNCTION; -- path_head_to_tail

FUNCTION path_reversed(
             a_path: path
    ): path;

  LOCAL
    the_reverse : path;
  END_LOCAL;
  IF 'METAL_CASTING.ORIENTED_PATH' IN TYPEOF(a_path) THEN
    the_reverse := oriented_path(a_path\oriented_path.path_element,NOT
        a_path\oriented_path.orientation);
  ELSE
    the_reverse := oriented_path(a_path,FALSE);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- path_reversed

FUNCTION scalar_times_vector(
```

```
                 scalar: REAL;
                 vec: vector_or_direction
        ): vector;

      LOCAL
        v      : direction;
        mag    : REAL;
        result : vector;
      END_LOCAL;
      IF (NOT EXISTS(scalar)) OR (NOT EXISTS(vec)) THEN
        result := ?;
      ELSE
        IF 'METAL_CASTING.VECTOR' IN TYPEOF(vec) THEN
          v := vec.orientation;
          mag := scalar * vec.magnitude;
        ELSE
          v := vec;
          mag := scalar;
        END_IF;
        IF mag < 0 THEN
          REPEAT i := 1 TO SIZEOF(v.direction_ratios) BY 1;
            v.direction_ratios[i] := -v.direction_ratios[i];
          END_REPEAT;
          mag := -mag;
        END_IF;
        result.orientation := normalise(v);
        result.magnitude := mag;
      END_IF;
      RETURN(result);

    END_FUNCTION; -- scalar_times_vector

    FUNCTION second_proj_axis(
                 z_axis, x_axis, arg: direction
        ): direction;

      LOCAL
        temp   : vector;
        v      : direction;
        y_axis : vector;
      END_LOCAL;
      IF NOT EXISTS(arg) THEN
        v := direction([0,1,0]);
      ELSE
        v := arg;
      END_IF;
      temp := scalar_times_vector(dot_product(v,z_axis),z_axis);
      y_axis := vector_difference(v,temp);
      temp := scalar_times_vector(dot_product(v,x_axis),x_axis);
      y_axis := vector_difference(y_axis,temp);
      y_axis := normalise(y_axis);
      RETURN(y_axis.orientation);
```

```
END_FUNCTION; -- second_proj_axis

FUNCTION set_of_topology_reversed(
             a_set: set_of_reversible_topology_item
    ): set_of_reversible_topology_item;

  LOCAL
    the_reverse : set_of_reversible_topology_item;
  END_LOCAL;
  the_reverse := [];
  REPEAT i := 1 TO SIZEOF(a_set) BY 1;
    the_reverse := the_reverse + topology_reversed(a_set[i]);
  END_REPEAT;
  RETURN(the_reverse);

END_FUNCTION; -- set_of_topology_reversed

FUNCTION shell_reversed(
             a_shell: shell
    ): shell;

  LOCAL
    the_reverse : shell;
  END_LOCAL;
  IF 'METAL_CASTING.ORIENTED_OPEN_SHELL' IN TYPEOF(a_shell) THEN
    the_reverse := oriented_open_shell(a_shell\oriented_open_shell.
        open_shell_element,NOT a_shell\oriented_open_shell.orientation);
  ELSE
    IF 'METAL_CASTING.OPEN_SHELL' IN TYPEOF(a_shell) THEN
      the_reverse := oriented_open_shell(a_shell,FALSE);
    ELSE
      IF 'METAL_CASTING.ORIENTED_CLOSED_SHELL' IN TYPEOF(a_shell) THEN
        the_reverse := oriented_closed_shell(a_shell\
            oriented_closed_shell.closed_shell_element,NOT a_shell\
            oriented_closed_shell.orientation);
      ELSE
        IF 'METAL_CASTING.CLOSED_SHELL' IN TYPEOF(a_shell) THEN
          the_reverse := oriented_closed_shell(a_shell,FALSE);
        ELSE
          the_reverse := ?;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- shell_reversed

FUNCTION surface_weights_positive(
             b: rational_b_spline_surface
    ): BOOLEAN;
```

```
  LOCAL
    result : BOOLEAN := TRUE;
  END_LOCAL;
  REPEAT i := 0 TO b.u_upper BY 1;
    REPEAT j := 0 TO b.v_upper BY 1;
      IF b.weights[i][j] <= 0 THEN
        result := FALSE;
        RETURN(result);
      END_IF;
    END_REPEAT;
  END_REPEAT;
  RETURN(result);

END_FUNCTION; -- surface_weights_positive

FUNCTION topology_reversed(
             an_item: reversible_topology
    ): reversible_topology;
  IF 'METAL_CASTING.EDGE' IN TYPEOF(an_item) THEN
    RETURN(edge_reversed(an_item));
  END_IF;
  IF 'METAL_CASTING.PATH' IN TYPEOF(an_item) THEN
    RETURN(path_reversed(an_item));
  END_IF;
  IF 'METAL_CASTING.FACE_BOUND' IN TYPEOF(an_item) THEN
    RETURN(face_bound_reversed(an_item));
  END_IF;
  IF 'METAL_CASTING.FACE' IN TYPEOF(an_item) THEN
    RETURN(face_reversed(an_item));
  END_IF;
  IF 'METAL_CASTING.SHELL' IN TYPEOF(an_item) THEN
    RETURN(shell_reversed(an_item));
  END_IF;
  IF 'SET' IN TYPEOF(an_item) THEN
    RETURN(set_of_topology_reversed(an_item));
  END_IF;
  IF 'LIST' IN TYPEOF(an_item) THEN
    RETURN(list_of_topology_reversed(an_item));
  END_IF;
  RETURN(?);

END_FUNCTION; -- topology_reversed

FUNCTION using_representations(
             item: representation_item
    ): SET OF representation;

  LOCAL
    results            : SET OF representation;
    i                  : INTEGER;
    intermediate_items : SET OF representation_item;
```

```
    result_bag         : BAG OF representation;
  END_LOCAL;
  result_bag := USEDIN(item,'METAL_CASTING.REPRESENTATION.ITEMS');
  IF SIZEOF(result_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(result_bag) BY 1;
      results := results + result_bag[i];
    END_REPEAT;
  END_IF;
  intermediate_items := QUERY ( z <* bag_to_set(USEDIN(item,'')) | (
      'METAL_CASTING.REPRESENTATION_ITEM' IN TYPEOF(z)) );
  IF SIZEOF(intermediate_items) > 0 THEN
    REPEAT i := 1 TO HIINDEX(intermediate_items) BY 1;
      results := results + using_representations(intermediate_items[i]);
    END_REPEAT;
  END_IF;
  RETURN(results);

END_FUNCTION; -- using_representations

FUNCTION valid_basis_curve_in_2d_wireframe(
             crv: curve;
             schma: STRING
    ): BOOLEAN;
  IF (schma + '.TRIMMED_CURVE') IN TYPEOF(crv) THEN
    RETURN(valid_basis_curve_in_2d_wireframe(crv\trimmed_curve.
        basis_curve,schma));
  ELSE
    IF (schma + '.COMPOSITE_CURVE') IN TYPEOF(crv) THEN
      RETURN(SIZEOF(QUERY ( ccs <* crv\composite_curve.segments | (NOT
          valid_basis_curve_in_2d_wireframe(ccs.parent_curve,schma)) )) =
           0);
    ELSE
      IF (schma + '.OFFSET_CURVE_2D') IN TYPEOF(crv) THEN
        RETURN(valid_basis_curve_in_2d_wireframe(crv\offset_curve_2d.
            basis_curve,schma));
      ELSE
        IF SIZEOF([schma + '.LINE',schma + '.B_SPLINE_CURVE',schma +
            '.CIRCLE',schma + '.ELLIPSE',schma + '.HYPERBOLA',schma +
            '.PARABOLA',schma + '.POLYLINE'] * TYPEOF(crv)) = 1 THEN
          RETURN(TRUE);
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- valid_basis_curve_in_2d_wireframe

FUNCTION valid_calendar_date(
             date: calendar_date
    ): LOGICAL;
  IF NOT ((1 <= date.day_component) AND (date.day_component <= 31))
```

```
      THEN
    RETURN(FALSE);
  END_IF;
  CASE date.month_component OF
     2 : BEGIN
           IF leap_year(date.year_component) THEN
             RETURN((1 <= date.day_component) AND (date.day_component <= 29));
           ELSE
             RETURN((1 <= date.day_component) AND (date.day_component <= 28));
           END_IF;
         END;
     4 : RETURN((1 <= date.day_component) AND (date.day_component <= 30));
     6 : RETURN((1 <= date.day_component) AND (date.day_component <= 30));
     9 : RETURN((1 <= date.day_component) AND (date.day_component <= 30));
    11 : RETURN((1 <= date.day_component) AND (date.day_component <= 30));
   OTHERWISE  :
      RETURN(TRUE);
   END_CASE;

END_FUNCTION; -- valid_calendar_date

FUNCTION valid_time(
             time: local_time
    ): BOOLEAN;
  IF EXISTS(time.second_component) THEN
    RETURN(EXISTS(time.minute_component));
  ELSE
    RETURN(TRUE);
  END_IF;

END_FUNCTION; -- valid_time

FUNCTION valid_units(
             m: measure_with_unit
    ): BOOLEAN;
  IF 'METAL_CASTING.LENGTH_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(1,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'METAL_CASTING.MASS_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,1,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'METAL_CASTING.TIME_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,1,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
```

```
END_IF;
IF 'METAL_CASTING.ELECTRIC_CURRENT_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,1,0,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'METAL_CASTING.THERMODYNAMIC_TEMPERATURE_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,1,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'METAL_CASTING.AMOUNT_OF_SUBSTANCE_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,0,1,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'METAL_CASTING.LUMINOUS_INTENSITY_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,0,0,1) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'METAL_CASTING.PLANE_ANGLE_MEASURE' IN TYPEOF(m.value_component)
     THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,0,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'METAL_CASTING.SOLID_ANGLE_MEASURE' IN TYPEOF(m.value_component)
     THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,0,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'METAL_CASTING.AREA_MEASURE' IN TYPEOF(m.value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(2,0,0,0,0,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'METAL_CASTING.VOLUME_MEASURE' IN TYPEOF(m.value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(3,0,0,0,0,0,0) THEN
```

```
        RETURN(FALSE);
      END_IF;
    END_IF;
    IF 'METAL_CASTING.RATIO_MEASURE' IN TYPEOF(m.value_component) THEN
      IF derive_dimensional_exponents(m.unit_component) <>
          dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN(FALSE);
      END_IF;
    END_IF;
    IF 'METAL_CASTING.POSITIVE_LENGTH_MEASURE' IN TYPEOF(m.value_component)
         THEN
      IF derive_dimensional_exponents(m.unit_component) <>
          dimensional_exponents(1,0,0,0,0,0,0) THEN
        RETURN(FALSE);
      END_IF;
    END_IF;
    IF 'METAL_CASTING.POSITIVE_PLANE_ANGLE_MEASURE' IN TYPEOF(m.
        value_component) THEN
      IF derive_dimensional_exponents(m.unit_component) <>
          dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN(FALSE);
      END_IF;
    END_IF;
    RETURN(TRUE);

  END_FUNCTION; -- valid_units

  FUNCTION vector_difference(
               arg1, arg2: vector_or_direction
      ): vector;

    LOCAL
      ndim   : INTEGER;
      mag2   : REAL;
      mag1   : REAL;
      mag    : REAL;
      res    : direction;
      vec1   : direction;
      vec2   : direction;
      result : vector;
    END_LOCAL;
    IF (NOT EXISTS(arg1)) OR (NOT EXISTS(arg2)) OR (arg1.dim <> arg2.dim)
         THEN
      result := ?;
    ELSE
      BEGIN
        IF 'METAL_CASTING.VECTOR' IN TYPEOF(arg1) THEN
          mag1 := arg1.magnitude;
          vec1 := arg1.orientation;
        ELSE
          mag1 := 1;
          vec1 := arg1;
```

```
        END_IF;
        IF 'METAL_CASTING.VECTOR' IN TYPEOF(arg2) THEN
          mag2 := arg2.magnitude;
          vec2 := arg2.orientation;
        ELSE
          mag2 := 1;
          vec2 := arg2;
        END_IF;
        vec1 := normalise(vec1);
        vec2 := normalise(vec2);
        ndim := SIZEOF(vec1.direction_ratios);
        mag := 0;
        REPEAT i := 1 TO ndim BY 1;
          res.direction_ratios[i] := (mag1 * vec1.direction_ratios[i]) - (
              mag2 * vec2.direction_ratios[i]);
          mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
        END_REPEAT;
        IF mag > 0 THEN
          result.magnitude := SQRT(mag);
          result.orientation := res;
        ELSE
          result.magnitude := 0;
          result.orientation := vec1;
        END_IF;
      END;
    END_IF;
    RETURN(result);

  END_FUNCTION; -- vector_difference

END_SCHEMA; -- metal_casting
(*
```

Annex B
(normative)

Short names of entities

Table B.1 provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

Table B.1 – Short names of entities

Entity names	Short names
ACTION	ACTION
ACTION_ASSIGNMENT	ACTASS
ACTION_DIRECTIVE	ACTDRC
ACTION_METHOD	ACTMTH
ACTION_REQUEST_ASSIGNMENT	ACRQAS
ACTION_REQUEST_SOLUTION	ACRQSL
ACTION_REQUEST_STATUS	ACRQST
ACTION_STATUS	ACTSTT
ADDRESS	ADDRSS
ADVANCED_BREP_SHAPE_REPRESENTATION	ABSR

Annex C
(normative)

Implementation method specific requirements

Conformance to this part of ISO 10303 shall be realized in one or more implementation methods. The implementation methods define what types of exchange behavior is required with respect to this part of ISO 10303. One implementation method is currently defined: ISO 10303-21.

For an exchange structure, the file format shall be encoded according to the syntax and EXPRESS language mapping defined in ISO 10303-21 and the AIM defined in Annex A of this part of ISO 10303. The header of the exchange structure shall identify use of this part of ISO 10303 by the schema name 'config_control_design'.

Annex D
(normative)

Protocol Implementation Conformance Statement (PICS) proforma

This clause lists the optional elements of this Part of ISO 10303. An implementation may chose to support any combination of these optional elements. However, certain combinations of options are likely to be implemented together. These combinations are called conformance classes and are described in the subclauses of this annex. This annex is in the form of a questionnaire. This questionnaire is intended to be filled out by the implementor and may be used in preparation for conformance testing by a testing laboratory. The completed PICS proforma is referred to as a PICS.

Annex E
(normative)

Information object registration

E.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(223) version(1) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

E.2 Schema identification

E.2.1 metal_casting expanded schema identification

To provide for unambiguous identification of the **metal_casting expanded schema** in an open information system, the object identifier

{ iso standard 10303 part(223) version(1) object(1) metal-casting expanded schema() }

is assigned to the **metal_casting expanded schema** schema (see). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

E.2.2 metal_casting short form schema identification

To provide for unambiguous identification of the **metal_casting short form schema** in an open information system, the object identifier

{ iso standard 10303 part(223) version(1) object(1) metal-casting short form schema() }

is assigned to the **metal_casting short form schema** schema (see). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex F
(informative)

Application activity model

The application activity model (AAM) is provided to aid in understanding the scope and information requirements defined in this application protocol. The model is presented as a set of definitions of the activities and the data and a set of activity figures. The AAM covers activitites that go beyond the scope of this application protocol. The viewpoint of the application activity model is from a design engineer. The purpose of the application activity model is to clarify the context and scope of this application protocol.

F.1 Application activity model definitions

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this application protocol. The definitions given in this annex do not supersede the definitions given in the main body of the text.

F.1.9 Cast a part: all activities and materials required to produce a casting.

F.1.10 Casting machine capabilities: characteristics of the available equipment which the casting processes must accommodate in order to make castings.

F.1.11 Casting part description: all information needed to describe and characterize the part or component which is to be cast, including the complete geometry of the component and relevant attributes, such as dimensional tolerances and any specified surface finishes, the mechanical properties required in the casting, the alloy to be used, and regulations, standards and specifications which apply. Information may also be included about special finishing or testing procedures required by the customer. If a sampling is required prior to commencement of full scale production, these requirements are also specified. Included are detailed descriptions of two items: the part originally by the customer, and a description of the casting part. The geometry of these two parts usually differ slightly. For example, the casting part would contain machining allowances.

F.1.12 Casting process data: the information about the processing methods and techniques used to produce the casting. It includes sand casting process data, die casting process data, investment casting process data, and finishing and inspection process data.

F.1.13 Casting for repair: a casting which has failed inspection tests, and is sent to be reworked or modified, so that it may afterwards be acceptable to the customer.

NOTE – Impregnation of castings which must be pressure tight, or welding of cracks are examples of actions which may be taken to repair castings.

F.1.14 Charge materials and additives for casting: all items used to produce the liquid metal used in casting, including ingot, recycled gates and risers, fluxes, alloying additives charged to the melting furnace, grain refiners, modifiers or innoculants, hardeners or other master alloys and fluxing salts or gases.

F.1.15 Core machine or core box: the core boxes and related equipment used to produce cores.

F.1.16 Core materials: materials used to produce cores, including bonded sand mixes, plaster in some applications, and steel or iron in die casting.

F.1.17 Core processing data: information related to the manufacture of cores, including the composition of materials used, the core box temperature and the holding time for production of shell cores, the curing cycle employed for bonded sand cores, the moisture content of the cores, and the results of any mechanical tests on cores.

F.1.18 Cores: a component added to the mold or die, which is used to produce undercuts or hollow sections.

> NOTE – Cores for low pressure casting processes are usually made from bonded sand, and are produced in a core box or core machine. A core for high pressure die casting is either fixed or movable, and is made from steel or cast iron. A movable core is usually called a slide. It slides into the mold cavity before pouring the shot, and is removed from the casting before the casting is extracted from the mold.

F.1.19 Drawing tools: hardware and software used in the design process to characterize the geometry, shape and size of the parts, or components, under consideration.

F.1.20 Finish: the raw casting is given one or more treatments in order to prepare it to meet the customer's specifications and requirements. The activities included in finishing are: trimming or cutting gates and risers from the casting, grinding off flash and other unwanted material, shot blasting or other surface finishing techniques, heat treatment, hipping, machining, cleaning, plating and painting.

F.1.21 Finish and inspect: the sequence of activities required to finish a raw casting for the customer, including trimming, machining, heat treating and inspection.

F.1.22 Finished casting: a raw casting which has received all necessary finishing operations, but which has not received final inspection and testing.

F.1.23 Finishing and inspection process data: information which characterizes the process parameters used in finishing operations; and the performance of the castings at each finishing operation; as well as the results of all test and inspection procedures used to evaluate the castings. The process data for repaired castings is also included.

F.1.24 Finishing equipment and heat treat furnaces: cutting and grinding equipment used to trim or remove rigging and flashing from the casting, equipment used to grind, machine or surface-condition castings, and furnaces used to heat treat or HIP the castings. Items such as quench tanks, heat treating fixtures and baskets are also included.

F.1.25 Finishing process data: information which characterizes the process parameters used in finishing operations, and the performance of the castings at each finishing operation. Included here are data on the heat treatment cycles and hipping procedures employed, and the number of castings scrapped at each finishing stage.

F.1.26 Foundry process specifications: in-house, or foundry, specifications used to characterize and regulate production processes employed in casting.

F.1.27 Furnace system: equipment used to produce the high temperatures required to melt metal prior to casting, or to produce the heat needed to melt out wax patterns in the dewaxing process.

F.1.28 Furnace system, crucibles, ladles, pumps, cranes and robots: equipment used to melt molten metal, prepare it for casting, and transfer the molten metal into the prepared sand mold, investment mold, or die.

F.1.29 Inspect and Test: subject the finished castings to the final test and inspection procedures specified for the casting.

> NOTE – Examples of inspection procedures are visual examinations for surface cracks and porosity; radiographic examinations for inclusions, porosity or shrinkage; ultrasonic inspection for internal defects; and checking that the casting meets the dimensional specifications. Examples of testing procedures are measurements of tensile or other mechanical properties, hardness measurements, breaking tests and checks for pressure tightness.

F.1.30 Inspected casting: the cast product, which has received the desired finishing operation, and which has been inspected and has been found to meet the customer's requirements. It is ready to be shipped to the customer from the foundry.

F.1.31 Inspection equipment: tools and machinery used to conduct inspection procedures.

F.1.32 Inspection process data: quality control process data which records the results of inspection and testing procedures that are performed on the finished casting, prior to shipping the casting to the customer.

> NOTE – The information included would be: results of visual inspection, compliance with dimensional tolerances, measurement of tensile properties, hardness measurements, microscopic examination, x-ray or other radiographic test results, measurement of pressure tightness, results of ultrasonic examination, and results of fluoroscopic or die penetrant tests for surface cracks and porosity.

F.1.33 Machine tools: equipment used to cut, drill, mill, grind and otherwise shape components in the casting process.

F.1.34 Machine tools, molding machines, core machines or core boxes: equipment used in production of sand molds and cores, and in machining of castings or die components.

F.1.35 Material specifications: items which describe the material under consideration, including its name or trade name, supplier or manufacturer, information related to its composition, and regulations, standards and specifications for this material.

F.1.36 Melt processing data and metal composition: quality control process data which characterizes the melting and melt treatment processes used to prepare liquid metal for casting, and the final resulting composition of the molten alloy.

NOTE – Items which may be included in metal processing data are the heat number, a list of materials and weights of materials charged to the furnace, furnace melting parameters, melt treatment parameters, any ladle treatments employed, and the results of any tests, including thermal analysis and measurements of inclusion or gas contents.

F.1.37 Mold temperature control system: system used to bring the die casting mold, or the investment casting mold, to the desired temperature prior to pouring liquid metal into the mold.

F.1.38 Molten metal: prepared molten metal, having the proper composition and temperature, before it is introduced into the mold.

F.1.39 Packaged casting: a casting which has been finished and inspected, and which has been packaged in the desired manner, and is ready for shipment to the customer.

F.1.40 Package: place of the finished and inspected cast part into the desired shipping container or containers, and weigh the castings before shipment.

NOTE – In some cases the customer specifies a procedure for packaging, or even the containers to be used. At times, finished castings are placed into specially designed, premolded skids or shipping containers which may be provided by the customer or by the foundry.

F.1.41 Packaging data: information related to the act of packaging, such as the number of finished cast parts, or the weight to be shipped.

F.1.42 Process modeling system: a computer-based simulation of the mold filling and metal freezing processes which occur during casting.

F.1.43 Process plan: a specified sequence of process operations, together with specifications regarding important process operating parameters to be used in each process step, which can be used to produce a single casting, or by repetition, a large quantity of castings. Included in the process plan would be any special instruction, as well as machine settings to be used in a process step.

F.1.44 Quality assurance data: detailed test and inspection results obtained from a sample lot of castings. These QA results are used to qualify the mold or die design for dimensional correctness, and to qualify the process plan as being capable of producing castings of sufficiently high quality.

F.1.45 Raw casting: the casting after it is removed from the mold, and before any trimming, grinding, finishing, machining, heat treating or inspection.

F.1.46 Regulations, standards and specifications: detailed and standardized specifications established by international, national, or trade organizations which apply to the material or process under consideration.

F.1.47 Repair: remove defects in castings, and restore the functionality of the cast part. Two examples of repair operations are welding, and impregnation of castings which must be pressure tight.

F.1.48 Repair process data: quality control process data which is used to characterize the repair procedures used for any particular casting.

F.1.49 Repaired casting: a defective casting which has received remedial action, and consequently is suitable for the customer's application.

F.1.50 Scrap to recycle: trim and scrapped castings which will be remelted in order to recover the contained metal.

F.1.51 Scrapped castings: castings which do not to meet the customer's specifications and requirements, and which cannot be repaired.

F.1.52 Secondary tooling design: engineering specifications for any of a number of secondary tooling devices used in finishing, inspection and packaging, when castings are produced in large quantities. The items described here would include extraction grippers and die spray nozzles in die casting, dies or fixtures for trim or straightening presses, machining fixtures, plating fixtures, finishing fixtures, specially designed shipping containers, gauging fixtures, heat treating fixtures, core removal fixtures, painting fixtures, and leak testing fixtures.

F.1.53 Shipping containers: especially designed, pre-molded skids or containers, which are used to hold and contain the finished cast parts during shipment. These may be provided by the customer or by the foundry.

F.1.54 Trim: unwanted material which is cut away from the desired portions of the raw casting. Trim includes gates, risers, runners, and flash on the raw casting.

F.1.55 Assemble patterns and rigging onto plate: mount the patterns and the rigging components onto the pattern plate.

F.1.56 Build patterns: produce the patterns required to make molds in sand casting.

F.1.57 Build pattern and rigging assembly: make the master of the pattern, if it is required, construct the desired patterns, fabricate the needed rigging components, and mount the patterns and the rigging components onto the pattern plate.

F.1.58 Build rigging: construct the rigging components required for the pattern and rigging assembly.

F.1.59 Build sand mold: the sequence of processes used to produce the completed mold in sand casting. This includes building the pattern and rigging assembly, making the sand mold, placing cores, filters and inserts into the sand mold, and inspecting and preparing the mold for casting.

> NOTE – The activity "Make sand mold", described later, is more limited in meaning, and is concerned only with the production of the empty cope and drag halves of the mold, by forming sand against the pattern and rigging assembly.

F.1.60 Cores and inserts design: the complete geometric description of the cores and inserts, their location in the sand mold, the specified core or insert materials, and the process specified for their production.

NOTE – A description of any masters used to create core molds is also included.

F.1.61 Data required for simulation of sand casting: a complete description of the geometry of the sand casting, rigging components and attributes which characterize the materials in contact with the metal at any particular surface of the casting.

NOTE – Attributes which characterize the materials would include a description of sand, chills, sleeves, filters and cores in the sand mold.

F.1.62 Description of master of pattern: information needed to characterize the design of the master which will be constructed during pattern manufacture. This includes the measured dimensions, the material to be used for its manufacture, and any identifications assigned to the master.

NOTE – The master is a scaled form, which is used to produce a sand casting pattern of the desired size and shape, often by machining while using the master as a template.

F.1.63 Description of masters of cores: information needed to characterize the design of masters. This includes the measured dimensions, the material to be used for their manufacture, and any identifications assigned to the master.

NOTE – The master is a scaled form, which is used to produce a core box of the desired size and shape, often by machining while using the master as a template.

F.1.64 Design cores and inserts for sand mold: specify the geometry of the cores and inserts, their location in the sand mold, and the materials to be used in their production.

F.1.65 Design pattern for sand mold: specify the complete geometric description of the pattern, the location of the pattern or patterns on the pattern plate, and the process to be used for the production of the patterns.

NOTE – The design pattern process includes describing the geometry of the master, selecting the parting plane or parting planes, deciding whether a single pattern or multiple pattern casting is desired, describing the geometry of the pattern or patterns on each side of the plate, and assigning the location of the patterns on the pattern plate.

F.1.66 Design rigging and secondary tooling for sand mold: specify the geometry of the components of the rigging, their location with respect to the pattern on the pressure or match plate, and the materials to be used for their construction. Establish the design of secondary tooling required for finishing of the raw sand casting, the geometry of each tooling item, and the materials to be used for their construction.

NOTE – The secondary tooling devices are used in finishing and inspection, and include fixtures for trim or straightening presses, machining fixtures, finishing fixtures, gauging fixtures, heat treating fixtures, core removal fixtures, painting fixtures, plating fixtures, shipping containers and leak testing fixtures.

F.1.67 Design sand mold: specify the complete design of the sand mold, which consists of the design of the pattern or patterns, the rigging, cores and inserts. The design of the mold flask and the pattern plate is also included here.

F.1.68 Design sand mold and plan process: specify the sand mold design, simulate the sand casting process, produce a small production run of sample castings, and establish the process plan to be used in large scale production of castings.

NOTE – The sample castings are analyzed and tested in detail to provide quality assurance data, which may be submitted to the customer in order to qualify the foundry's mold design and possibly their process plan for subsequent large scale production of the casting part.

F.1.69 Filled sand mold: the sand mold and its contents immediately after molten metal has been poured into it, and before the metal has solidified and cooled to the point where it can be shaken out of the mold.

F.1.70 Inspect and prepare mold: the sand mold is examined for any defects, such as loose sand or cracks, and for dimensional tolerances.

NOTE – In processes which use bonded sand or plaster, the mold is allowed to harden or cure. If mold coatings are to be used, they are applied at this time. Cores, inserts and filters are set into the mold, and their placement is checked. The mold is given a final inspection and the mold is closed. Weights are placed on top of the mold, and brackets may also be placed at the top or sides of the mold. The mold is now ready to receive liquid metal.

F.1.71 Make cores: manufacture cores for sand casting.

F.1.72 Make masters: build core or pattern masters.

F.1.73 Make sand mold: manufacture the mold cavity used in sand casting.

NOTE – The mold consists of a hollow cavity, which is constructed by forming molding sand against both sides of the pattern and rigging assembly or pressure plate, removing the pattern, and the reassembling the top and bottom halves of the sand mold. This activity also includes the placement of any riser sleeves, chills or filters into the mold.

F.1.74 Master of cores: a scaled form used to produce a core box, often by machining while using the master as a template.

F.1.75 Master of pattern: a scaled form of the pattern used to produce the pattern, by casting impressions of the master, or by machining while using the master as a template.

F.1.76 Materials for masters: the materials used to produce core or pattern masters.

F.1.77 Materials for pattern, rigging, mold, inserts and cores for sand casting: the materials used to construct the pattern, the rigging, the mold, inserts and cores in sand casting.

F.1.78 Melt, pour, cool and shakeout: prepare liquid metal alloy, pour it into the sand mold, allow time for the sand casting to freeze and cool, and remove or shake out the raw sand casting from the sand mold.

F.1.79 Molding machines: the equipment used to make sand molds.

F.1.80 Patterns: shaped component used to produce molds in sand casting.

NOTE – There are three types of pattern or pattern plate assemblies. The first type is called a pressure plate. The pressure plate is usually made of one piece, by casting a plate together with impressions made from the pattern master. The pressure plate can also be made by machining. The second type of pattern is called a match plate. In this case master pieces, or copies of the master, are bolted or screwed onto the pattern plate. The third type of pattern is a loose pattern, where the pattern pieces are placed on a board, but not fixed permanently thereon. Loose patterns are normally employed when only a few castings are needed.

F.1.81 Pattern and rigging assembly: the completed assembly, used to make sand molds, in which the pattern or patterns and the rigging have been mounted or placed on the pattern plate.

F.1.82 Pattern and rigging materials: the materials used to construct the pattern or patterns and the rigging in sand casting.

F.1.83 Pattern design: information which characterizes the design of the sand casting patterns.

NOTE – The information includes a complete geometric description of the master used to make the pattern, the location of patterns on the pattern plate, the material to be used in constructing the patterns, the size of the mold–cope and drag heights– and the pattern plate size.

F.1.84 Pattern materials: the materials used to construct the sand casting patterns.

NOTE – Wood, plastic and aluminum are commonly used materials.

F.1.85 Place cores and inserts into mold: arrange cores and inserts inside the sand mold.

F.1.86 Pour and cool sand casting: pump or ladle liquid metal into the sand mold, and allow time for the casting to solidify and cool.

F.1.87 Pouring parameters: quality control process data which record the pouring process information for a casting.

NOTE – The parameters include the heat number, the temperature of the metal as it is poured into the mold, and the time to fill a mold during pouring.

F.1.88 Preliminary sand mold assembly: the sand mold assembly before final inspection and any curing of the mold.

NOTE – The preliminary sand mold assembly consists of the assembled cope and drag sections of the mold, inserted cores, inserts and chills, and any other specified rigging components.

F.1.89 Prepare melt for sand casting: charge materials are melted, refined, held at desired temperature and brought to the desired alloy composition in preparation for pouring into the sand mold.

F.1.90 Prepared sand mold: the assembled sand mold is ready to receive liquid metal.

NOTE – The prepared sand mold contains the needed cores, chills, inserts and rigging components, has been inspected; and in case of bonded sand or plaster mold casting, it has also been cured.

F.1.91 Raw sand casting: the sand casting after it has been removed from the mold, and before finishing or inspection operations.

F.1.92 Recycled sand: sand that is recovered during shakeout of the casting, and which is recycled back into the foundry sand system.

F.1.93 Rigging: parts of the sand mold, aside from the pattern or patterns.

NOTE – The rigging thus includes items used to convey the molten metal in the mold such as the pouring cup, sprue, basins, gates, runners, risers and vents as well as other miscellaneous mold components such as chaplets or nails, internal and external chills, test bars, insulating sleeves, in-mold filters, and in-mold additions of flux, grain refiner or innoculant.

F.1.94 Rigging design: the complete geometric description of the rigging components, including their location on the pressure plate or in the mold, and a description of the materials to be used in construction of the rigging components.

F.1.95 Rigging materials: materials used to construct the rigging components, which are mounted or placed on the pattern plate in sand casting.

F.1.96 Sample and plan sand casting process: produce a small sample lot of castings, test and inspect the castings, and use the test results to establish an effective process plan, which can be used to produce quality castings in large quantities.

F.1.97 Sand: includes all materials used to make molds from a non-consumable pattern and rigging assembly; including the usual "green" sands; which are usually a mixture of silica or zircon sand, clay and water; bonded sands, and plaster.

NOTE – When a consumable wax pattern is employed to produce a plaster mold, the casting process is categorized under investment casting, not sand casting.

F.1.98 Sand and materials for mold: the sand mixture and other materials used to construct the mold in sand casting.

F.1.99 Sand cast a part: the sequence of activities and materials required to manufacture a product by sand casting.

F.1.100 Sand casting process data: information used to characterize the design and manufacture of the pattern and rigging, and the molding, melting and pouring processes used to produce a sand casting.

F.1.101 Sand mold: the hollow cavity constructed by forming the molding sand against both sides of the pattern and pattern plate.

F.1.102 Sand mold design: the information which characterizes the design of the sand casting pattern, the cores, inserts and the rigging; including information regarding the location of the components with respect to the parting plane, and the location of the components on the pattern plate.

F.1.103 Sand mold inspection and preparation results: information regarding the results of the final inspection and preparation of the sand mold.

F.1.104 Sand mold quality control data: quality control process data, which describe the processing of the sand mold, the cores, and inspection of the sand mold.

F.1.105 Sand mold processing data: quality control process data which characterizes the properties of the sand used in construction of the mold.

NOTE – This processing data includes the measured composition of the sand mix, the results of tests to determine sand strength, compactability, permeability, and the measured grain size of the sand. Process Information about the sand mulling and sand molding operations is also included in sand mold processing data.

F.1.106 Sand mold quality control data: quality control process data which characterizes the properties of the sand used in construction of the mold, the core processing data, and the sand mold inspection and preparation results.

F.1.107 Sand recycle equipment: equipment used to recover used sand in the foundry.

NOTE – Sand heat exchangers and screening equipment fall into this category.

F.1.108 Shake out sand casting: after cooling, the filled sand mold is placed into a machine, which shakes the casting in order to separate it from the sand in the mold and cores.

F.1.109 Shakeout machine: equipment used to remove the sand from raw casting or knock out a casting.

F.1.110 Shakeout machine and sand recycle equipment: equipment used to shake out a casting, and to recycle the sand recovered in the shakeout process.

F.1.111 Simulate sand casting: model the sand casting process, by using a computer-based simulation of the mold filling and metal freezing processes which occur during sand casting.

F.1.112 Simulation results for sand casting: predictions of casting quality made by using a computer-based model of the mold filling and metal freezing processes which occur during sand casting.

NOTE – Typical sand casting simulation results would include predictions related to casting quality, such as the possible location of hot spots or shrinkage cavities, and the distribution of porosity; and may be in a text or pictorial format.

F.1.113 Used sand: the sand resulting from the shakeout process.

F.1.114 Waste sand: sand which is not of the proper size, or is otherwise unsuitable, to be used for the molding material in sand casting.

F.1.115 Build die: produce and assemble the finished die mold assembly in the die casting machine.

F.1.116 Build die components: manufacture the die components by casting, machining or hipping, to bring them to the final desired shape, dimensions and surface finish.

F.1.117 Cool and extract: allow the molten metal in the die to solidify and cool, and then extract or eject the solidified casting from the die.

F.1.118 Data required for simulation of die casting: a complete description of the geometry of the die casting and the die mold, proposed location of any cooling channels, attributes which characterize the materials in contact with the metal at any particular surface of the casting.

NOTE – Attributes which characterize the materials would include a description of chills, sleeves, filters and mold coatings in the die.

F.1.119 Design die: consists of designing the layout of die sub assemblies in the die, and designing all sub assemblies, components, and the die rigging.

F.1.120 Design die and plan die casting process: specify the die design, simulate the die casting process, produce a small production run of sample castings, and establish the process plan to be used in large scale production of castings.

NOTE – The sample castings are analyzed and tested in detail to provide quality assurance data, which may be submitted to the customer in order to qualify the foundry's die design and their process plan for subsequent large scale production of the casting part.

F.1.121 Design die layout: establish the overall design and layout of the complete die mold assembly.

NOTE – This design includes a list of all die sub-assemblies needed for the die mold, layout and assembly drawings, and material or stock lists of needed components and sub assemblies.

F.1.122 Design die rigging and secondary tooling: specify the die rigging design and the secondary tooling design.

F.1.123 Design die sub-assemblies: establish the design of die mold sub-assemblies required for the die mold.

NOTE – Sub-assemblies would consist of items such as: fixed cores, slides, consumable cores, inserts, core pin layouts, ejector pin and plate assemblies, coolant layouts, hydraulic layouts, electrical layouts and safety switch layouts.

F.1.124 Die assembly: consists of all individual die components assembled together.

F.1.125 Die assembly data: production information related to the act of constructing the die assembly.

NOTE – Information included in die assembly data is the name and identification number of the assembly, personnel responsible for assembly, and inspection results.

F.1.126 Die assembly design: completed design of the entire die mold assembly, which includes the die layout, the die sub-assemblies design and the die rigging design.

F.1.127 Die cast a part: sequence of activities and materials required to manufacture a product by die casting.

F.1.128 Die casting machine: equipment which houses the die assembly and produces die castings.

F.1.129 Die casting process data: information used to characterize the design and manufacture of the die, and the processes employed to produce a die casting.

F.1.130 Die components: the individual cast or machined components, before any heat treatment, which together are used to produce the final die assembly in the die casting machine.

NOTE – Die components include items such as mold sections, cores or slides, the shot sleeve and ejector pins.

F.1.131 Die layout: specifies the overall layout of the die assembly, and how the part to be cast is placed inside the die mold.

F.1.132 Die or mold temperature control system: the machinery and controls used to cool or heat the mold in die casting, thereby bringing it to the desired temperature for pouring the casting.

F.1.133 Die preparation process data: describes the processes and procedures used to prepare the die before casting begins, including data on the measured temperature of the mold and on the application of mold coatings.

F.1.134 Die property data: relevant information regarding the construction and assembly of the die mold used in die casting.

NOTE – Die property data include data on the finished die components, the heat treatment of die components, the construction of the die assembly, and the process of mounting the die assembly on the die casting machine.

F.1.135 Die rigging design: the design and geometric description of the die rigging, which includes the size and material of the shot sleeve, location of the sprue hole, and the size and location of the runners, gates, overflows, risers and vents.

NOTE – In a sense, the die rigging may be considered to be a special sub-assembly.

F.1.136 Die sub-assemblies design: a complete description of the geometry or other design of die sub-assemblies, together with a specification of the materials to be used in construction of the sub-assemblies.

NOTE – Typical sub-assemblies are: fixed cores, slides, consumable cores, inserts, core pin layouts, ejector pin and plate assemblies, coolant layouts, hydraulic layouts, electrical layouts, and safety switch layouts.

F.1.137 Die temperature control system: the machinery and controls used to cool or heat the mold or die in die casting; thereby bringing it to the desired temperature for pouring the casting.

F.1.138 Filled die mold: the die mold and its contents immediately after molten metal has been shot into it, and before the metal has solidified and cooled to the point where it can be extracted.

F.1.139 Finished die components data: information related to the production of die components used to construct the die assembly.

NOTE – Finished die components data include the name of the component, the supplier or manufacturer, and relevant information about the materials used to manufacture the component.

F.1.140 Heat treat data: quality control process data related to the heat treatment of individual die mold components.

NOTE – Examples of heat treat data are the times and temperatures employed in the heat treatment process, and the results of hardness or toughness measurements.

F.1.141 Heat treat die components: subject the specified die components to a sequence of high temperature treatments in order to produce the desired mechanical properties in the component.

NOTE – Heat treatment of tool steel die components for high pressure die casting would normally consist of a high temperature austenization treatment, followed by quenching and possibly tempering or stress relieving. Die components for low pressure die casting generally do not receive a heat treatment.

F.1.142 Heat treated die components: die components after they have received the specified heat treatment.

F.1.143 Materials for cores and die coatings: the materials used to produce cores, and to coat the die or permanent mold in die casting.

F.1.144 Materials for cores, die coatings, and die components: the materials used to produce accessory components for the die in die casting.

F.1.145 Materials for die components: the materials used to construct individual die components in die casting.

F.1.146 Melt, pour, cool and extract: prepare liquid metal alloy, pour it into the die mold, allow time for the die casting to freeze and cool, and then extract the casting from the die mold.

F.1.147 Mold temperature control system: used to bring the die casting mold to the desired temperature prior to pouring liquid metal into the mold.

F.1.148 Mount die assembly in machine: fix into position the completed die assembly into the die casting machine.

NOTE – At times the die assembly occurs directly on the die casting machine. In this case the die assembly and die mounting processes are combined.

F.1.149 Mounted die assembly: the completed die assembly, after it has been mounted on the die casting machine.

F.1.150 Mounted die assembly data: relevant manufacturing information regarding the location of the mounted die assembly on the die casting machine, which includes die assembly identification number and the die casting machine number.

F.1.151 Pour shot: inject, pump or ladle molten metal into the die mold.

NOTE – In the case of high pressure die casting this activity includes the use of a ram to force molten metal into the mold.

F.1.152 Prepare die: apply lubricant or die coatings and heat the die to the desired temperature.

F.1.153 Prepare melt for die casting: charge materials are melted, refined, held at desired temperature and brought to the desired alloy composition in preparation for pouring into the die mold.

F.1.154 Prepared die: the die which has been heated to the proper temperature and coated with suitable mold coatings and made ready to receive liquid metal for the first time.

F.1.155 Process data on cycle time: the information that describes the time between shots in die casting.

F.1.156 Raw die casting: the die casting after it is removed from the mold, and before any trimming, grinding, finishing, machining, heat treating or inspection.

F.1.157 Sample and plan die casting process: produce a small sample lot of castings, test and inspect the castings, and use the test results to establish an effective process plan, which can be used to produce quality castings in large quantities.

F.1.158 Scrapped castings: castings which do not to meet the customer's specifications and requirements, and which cannot be repaired.

F.1.159 Shot data: quality control process data recorded during the introduction of molten metal into the die casting mold.

NOTE – In high pressure die casting, a detailed profile of the ram velocity is usually recorded together with the pressure at end of fill and the intensification pressure. In gravity pour die casting, the fill time is measured. In tilt-pour permanent mold casting, the tilt rate and tilt profile is recorded.

F.1.160 Shot process data: quality control information describing the shot data, the die process data, and process data on the cycle time.

F.1.161 Shot process design parameters: the detailed design characterization of how liquid metal shall be introduced into the die mold, and the desired metal pouring temperature.

NOTE – In high pressure die casting the shot process parameters specify the ram speed profile and the ram intensification pressure. In low pressure or gravity pour die casting, the shot process parameters specify the fill time. In tilt-pour die casting, the shot process parameters specify the tilt rate of the mold.

F.1.162 Simulate die casting: model the die casting process, by using a computer-based simulation of the mold filling and metal freezing processes which occur during die casting.

F.1.163 Simulation results for die casting: predictions of casting quality made by using a computer-based model of the mold filling and metal freezing processes which occur during die casting.

NOTE – Typical die casting simulation results would include predictions related to casting quality, such as the possible location of hot spots or shrinkage cavities, and the distribution of porosity; and may be in a text or pictorial format. The die casting simulation results would include any predictions of die distortion or cracking, and estimates of the cooling needed at various points in the die.

F.1.164 Build investment pattern: produce the consumable pattern, which is covered with a ceramic mold in the investment casting processes.

NOTE – The pattern is normally built by injecting wax or foam into the pattern die.

F.1.165 Build investment pattern assembly: construct the investment pattern assembly, which consists of mounting the pattern or patterns onto the rigging assembly.

NOTE – The investment rigging assembly includes the pouring cup, down sprue, gates, runners and risers.

F.1.166 Build plaster mold: produce a plaster mold, by pouring plaster around the investment pattern assembly and allowing the plaster to harden and cure.

F.1.167 Build wax or foam pattern: produce the consumable wax or foam pattern, which is used to make ceramic molds in the investment casting processes.

F.1.168 Compaction process data for investment casting: quality control process data which records the information related to the sand compaction process, in lost foam investment casting.

NOTE – This should not be confused with the compactability test commonly used in sand casting, to characterize the quality of the molding sand.

F.1.169 Cores and investment pattern die: cores are components added to the wax pattern before investment to produce hollow sections in the casting. The pattern die is injected with wax or foam, to produce consumable patterns.

F.1.170 Cure shell or plaster mold: subject the dewaxed investment mold to a high temperature heat treatment to improve the strength of the ceramic shell or plaster mold and remove any remaining wax or moisture present in the mold.

F.1.171 Curing process data: quality control process data recorded about the mold curing process in investment casting.

NOTE – The curing process data includes the times and temperatures used in the curing furnace.

F.1.172 Data required for simulation of investment casting: a complete description of the geometry of the complete investment casting, or pattern assembly. Also included would be a description of the mold, and the temperature of the mold at pouring.

F.1.173 Design investment mold: specify the complete geometric description of the investment pattern, the design of components to be used to build the investment pattern assembly rigging, the location and orientation of patterns on the pattern assembly, and the design of any investment cores required. The processes to be used for the production of the components is also specified at this time. Any secondary tooling required to process the raw investment casting is also established.

F.1.174 Design investment pattern and pattern die: specify the complete geometric description of the investment pattern die, and any process parameters to be used for the production of patterns from the die. The geometry of the consumable pattern is also established at this time. This design activity also includes the development of any masters which may be used to produce investment pattern dies.

NOTE – Wax or foam is injected into the pattern die to produce investment patterns.

F.1.175 Design investment pattern assembly and plan process: specify the investment pattern assembly design, simulate the investment casting process, produce a small production run of sample castings, and establish the process plan to be used in large scale production of castings.

NOTE – The sample castings are analyzed and tested in detail to provide quality assurance data, which may be submitted to the customer in order to qualify the foundry's mold design and possibly their process plan for subsequent large scale production of the casting part.

F.1.176 Design pattern assembly: specify the complete geometric description of the components of the investment pattern assembly, and processes to be used for their production.

NOTE – The investment pattern assembly includes the pouring cup, down sprue, gates, runners and risers. Also specified in this design activity is the placement of the pattern or patterns on the rigging assembly.

F.1.177 Dewax: remove the wax pattern from the shell or plaster mold, by heating the green plaster mold or wax shell mold in a furnace or autoclave.

F.1.178 Dewaxing process data: quality control process data about the process used to remove the wax from a green plaster mold, or a wax shell mold, in investment casting.

NOTE – The dewaxing process data would include the steam pressure and time to reach pressure in an autoclave, or furnace temperature and treatment time in the flash firing process.

F.1.179 Die or mold temperature control system: the machinery and controls used to cool or heat the ceramic mold in investment casting, thereby bringing it to the desired temperature for pouring the casting.

F.1.180 Filled investment mold: The investment mold and its contents immediately after molten metal has been poured into it, and before the metal has solidified and cooled to the point where it can be knocked out of the mold.

F.1.181 Foam investment: the foam pattern assembly, after it has been invested with a ceramic shell, but before it is packed in sand.

NOTE – Foam investment is a green shell mold made from a foam pattern assembly.

F.1.182 Green plaster mold: the investment pattern assembly after it has been invested with plaster, but before it has been dewaxed or subjected to a high temperature curing process.

NOTE – Wax is commonly used to construct the pattern assembly for plaster mold investment casting.

F.1.183 Green shell mold: the wax or foam pattern assembly, immediately after it has been invested with a ceramic shell.

F.1.184 Heat shell or plaster mold: place the plaster mold or shell mold in a furnace, in order to bring it to the desired temperature to pour an investment casting.

F.1.185 Heated shell or plaster mold: the shell or plaster mold in investment casting, after it has been heated to the desired temperature for casting, and before it has received any liquid metal.

F.1.186 Inspect investment pattern assembly: examine the final pattern assembly or pattern tree for dimensional tolerance and completeness before investment.

F.1.187 Invest pattern: coat or invest the foam or wax pattern assembly with a ceramic shell or a plaster mold, dewax and cure the mold and; in the case of lost foam casting, compact sand around the mold.

F.1.188 Invest with ceramic shell: coat or invest a foam or wax investment pattern assembly with a ceramic shell mold coating.

NOTE – The shell mold coating usually consists of 6-8 thin layers of ceramic backed with a stucco coating.

F.1.189 Investment cast a part: the sequence of activities and materials required to manufacture a product by investment casting.

F.1.190 Investment casting mold design: the design of the complete investment mold assembly, which consists of the investment pattern, investment pattern dies used to produce patterns, the investment pattern assembly, and investment cores.

F.1.191 Investment casting process data: information used to characterize the design and manufacture of investment patterns and investment pattern assemblies, and the processes employed to produce an investment casting.

F.1.192 Investment cores design: a complete description of the geometry of cores used in investment casting, the materials used in their construction, and specifications regarding the procedures used to manufacture the cores.

F.1.193 Investment data: quality control process data recorded about the process of investment casting.

> NOTE – This information includes data on the manufacturing of plaster and ceramic shell investment molds, and on the dewaxing and curing of molds. Also included is data on the sand compaction process in lost foam investment casting.

F.1.194 Investment masters design: a complete geometric description of any masters used in investment casting. Masters might be used to produce pattern dies, or cores.

F.1.195 Investment materials: the materials used to invest the pattern and rigging assembly.

> NOTE – The materials most commonly used to build ceramic investment molds are colloidal silica gel-, ethyl silicate-, and gypsum-bonded ceramics.

F.1.196 Investment pattern: a consumable pattern, constructed of wax, plastic or foam, which has the same shape as the final desired cast product, but with slightly different dimensions to allow for thermal contraction of the casting after solidification. The pattern is invested with ceramic to produce part of the mold used in investment casting.

F.1.197 Investment pattern assembly: the completed wax or foam pattern assembly before investment. This includes the investment pattern or patterns and the investment rigging assembly. The investment rigging assembly includes the pouring cup, down sprue, gates, runners and risers.

> NOTE – The investment pattern assembly is often called a pattern tree and is constructed from wax, plastic or foam.

F.1.198 Investment pattern assembly data: the quality control process data related to the production of the finished investment casting pattern assembly.

> NOTE – This information includes data which characterizes the processes used to produce patterns, by injecting wax or foam into pattern dies, and the results of inspecting the final investment pattern assembly.

F.1.199 Investment pattern assembly design: the complete geometric description of the investment rigging components; including the pour cup, down sprue, gates, runners and risers; and the location and orientation of rigging components in the final pattern assembly.

> NOTE – The location of patterns in the final pattern assembly, or pattern tree, is also specified at this time. Also included are the processes specified for production of rigging components, and the desired materials to be used in the investment pattern assembly.

F.1.200 Investment pattern assembly inspection results: quality control process data related to the inspection of the preliminary investment pattern assembly.

F.1.201 Investment pattern design: the complete geometric description of the investment pattern, and the process specified for pattern production.

F.1.202 Investment pattern die design: the complete geometric description of dies used to create the wax or foam, and the materials to be used in their construction.

F.1.203 Investment pattern process data: quality control process data related to the production of investment casting patterns.

NOTE – The information includes die injection pressures used, cycle times, and the temperature of the injected wax or foam. In production of foam patterns this includes the vacuum applied during the pre-expansion of the beads, and the pattern molding parameters such as steam temperature, time of steam application, die cycle time and pattern aging time.

F.1.204 Knockout investment casting: place the cooled investment mold into a machine, which shakes or vibrates the mold, in order to break up the mold and separate it from the casting.

NOTE – Any washing procedures would also be included in this activity.

F.1.205 Lost foam mold: the mold produced for lost foam casting by investing a foam pattern assembly, and compacting sand around the foam investment.

F.1.206 Materials for wax or foam pattern and investment: The materials used to construct the consumable pattern, and the ceramic investment or mold materials in investment casting.

NOTE – Wax and foam and foam are commonly used to produce investment patterns, although other materials may be used in special applications.

F.1.207 Melt, pour, cool and knockout: prepare liquid metal alloy, pour the metal into the investment casting mold, allow time for the casting to freeze and cool, and remove or knock out the casting from the ceramic mold.

F.1.208 Plaster investment process data: quality control process data which records the processes used to manufacture the plaster mold, in investment casting.

NOTE – This information would contain the composition of the plaster, vibration and vacuum applied, if any, during investment and the time allowed for plaster to harden.

F.1.209 Pour and cool investment casting: pump or ladle liquid metal into the investment mold and allow time for the casting to solidify and cool.

F.1.210 Pouring parameters: quality control process data which record the pouring process information for a casting.

NOTE – This data includes the heat number, the temperature of the metal as it is poured into the mold, and the time to fill a mold during pouring. In investment casting it also includes the mold temperature prior to pouring.

F.1.211 Preliminary investment pattern assembly: The investment pattern assembly before inspection.

F.1.212 Prepare melt for investment casting: charge materials are melted, refined, held at desired temperature and brought to the desired alloy composition in preparation for pouring into the investment mold.

F.1.213 Raw investment casting: the investment casting after it is removed from the mold, and before any finishing or inspection.

F.1.214 Recycled wax: wax which is recovered during the dewaxing process, and which is reused in pattern or rigging assembly components.

F.1.215 Sample and plan investment casting process: Produce a small sample lot of castings, test and inspect the castings, and use the test results to establish an effective process plan, which can be used to produce quality castings in large quantities.

F.1.216 Sand compaction for lost foam compaction: the process of placing loose sand around the foam investment, and vibrating the sand to compact it around the lost foam mold.

F.1.217 Sand for lost foam compaction: loose sand, which is compacted around the lost foam mold in order to hold and support the mold during the pouring process.

F.1.218 Shakeout machine: equipment used to shake out or knock out a casting.

F.1.219 Shell coating process data: information recorded about the manufacture of the shell coating in investment casting.

NOTE – This information includes the composition of material used in each layer of the shell, any stucco materials applied, and drying times and drying conditions between layers.

F.1.220 Shell mold or plaster mold: two types of molds used in investment casting, which are dewaxed and cured before casting.

NOTE – The shell mold is constructed by applying a number of thin ceramic coatings to the investment pattern assembly. The plaster mold is constructed by immersing the wax investment pattern assembly into a container filled with freshly mixed plaster, and allowing the plaster to harden.

As used here plaster includes both gypsum bonded ceramics and other comparable ceramic slurries.

F.1.221 Simulate investment casting: model the investment casting process, by using a computer-based simulation of the mold filling and metal freezing processes which occur during investment casting.

F.1.222 Simulation results for investment casting: predictions of casting quality made by using a computer-based model of the mold filling and metal freezing processes which occur during investment casting.

NOTE – Typical investment casting results would include predictions related to casting quality, such as the possible location of hot spots or shrinkage cavities, and the distribution of porosity; and may be in a text or pictorial format.

F.1.223 Uncured plaster or shell mold: the plaster or ceramic shell mold, after dewaxing and before the high temperature mold curing process.

F.1.224 Wax or foam: the wax or foam materials used to construct the pattern assembly in investment casting.

F.1.225 Wax shell mold: a mold in investment casting, which has been made by constructing a wax pattern assembly, and investing the assembly with a ceramic shell coating.

NOTE – Wax shell mold is a green shell mold made from a wax pattern assembly.

F.2 Application activity model figures

The application activity model is given in figures F.2 through F.21. The graphical form of the application activity model is presented in the IDEF0 activity modelling format. Activities and data flows that are out of scope are marked with asterisks.

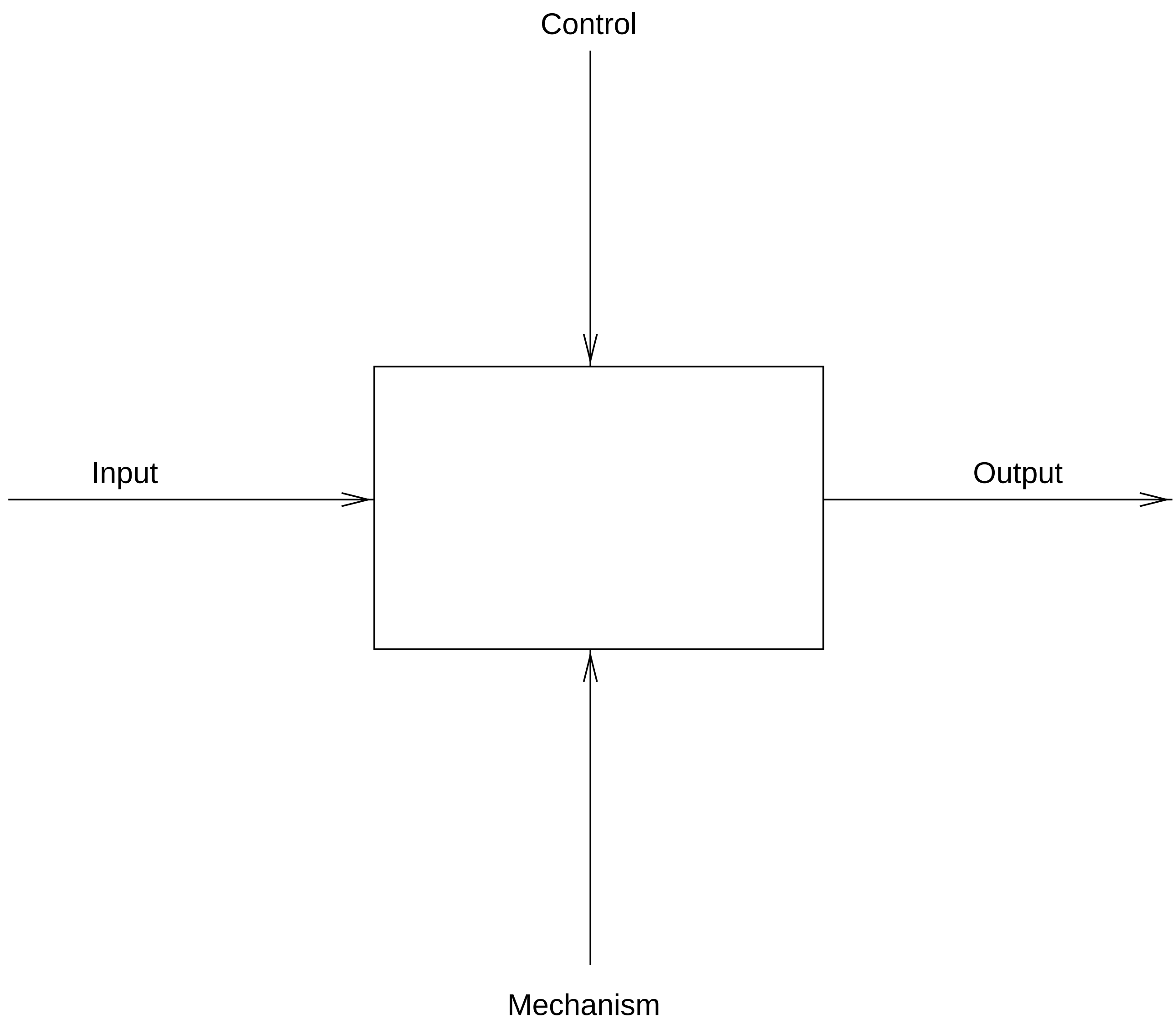


Figure F.1 – IDEF0 basic notation

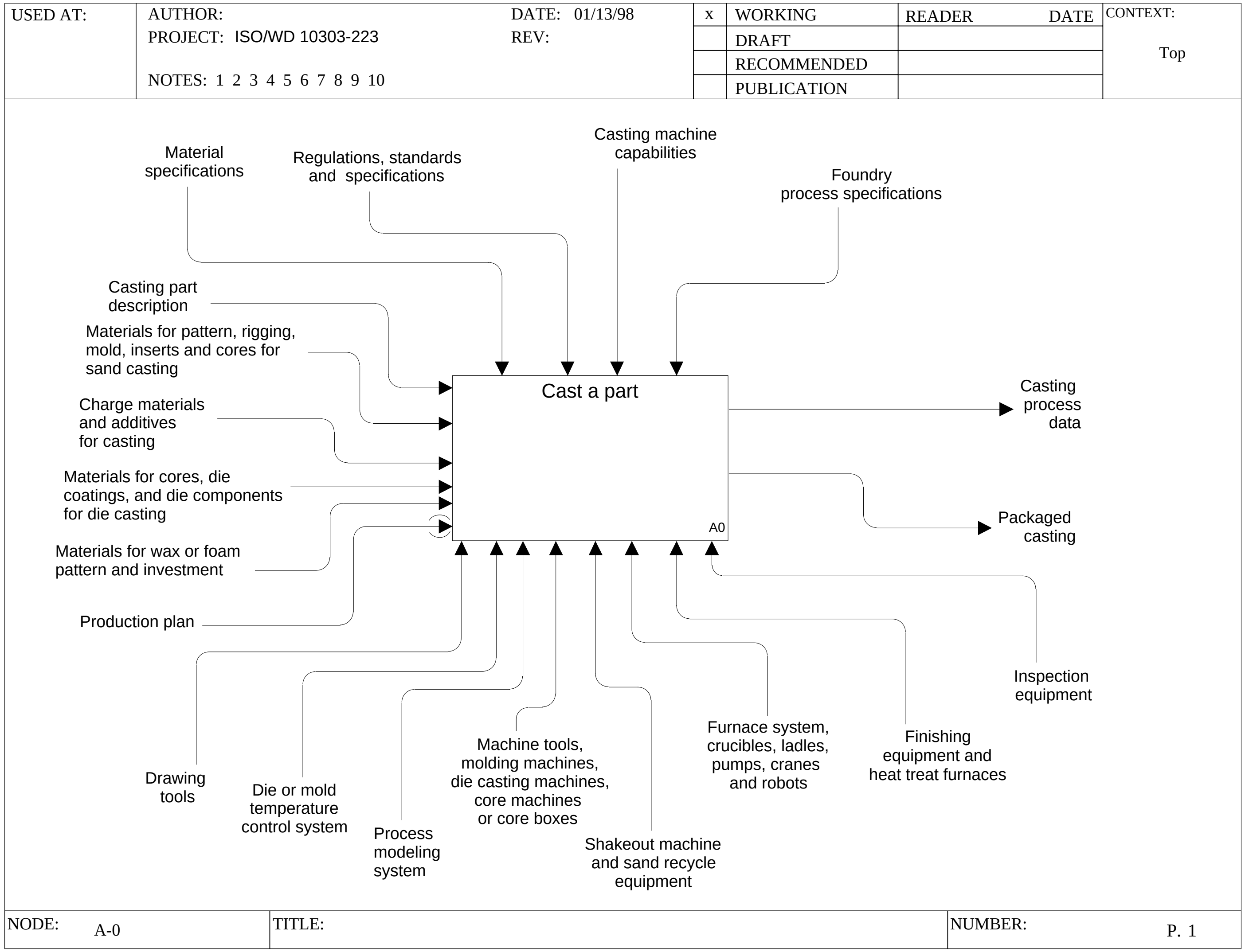


Figure F.2 – A-0 Exchange of Design and Manufacturing Product Information for Cast Parts

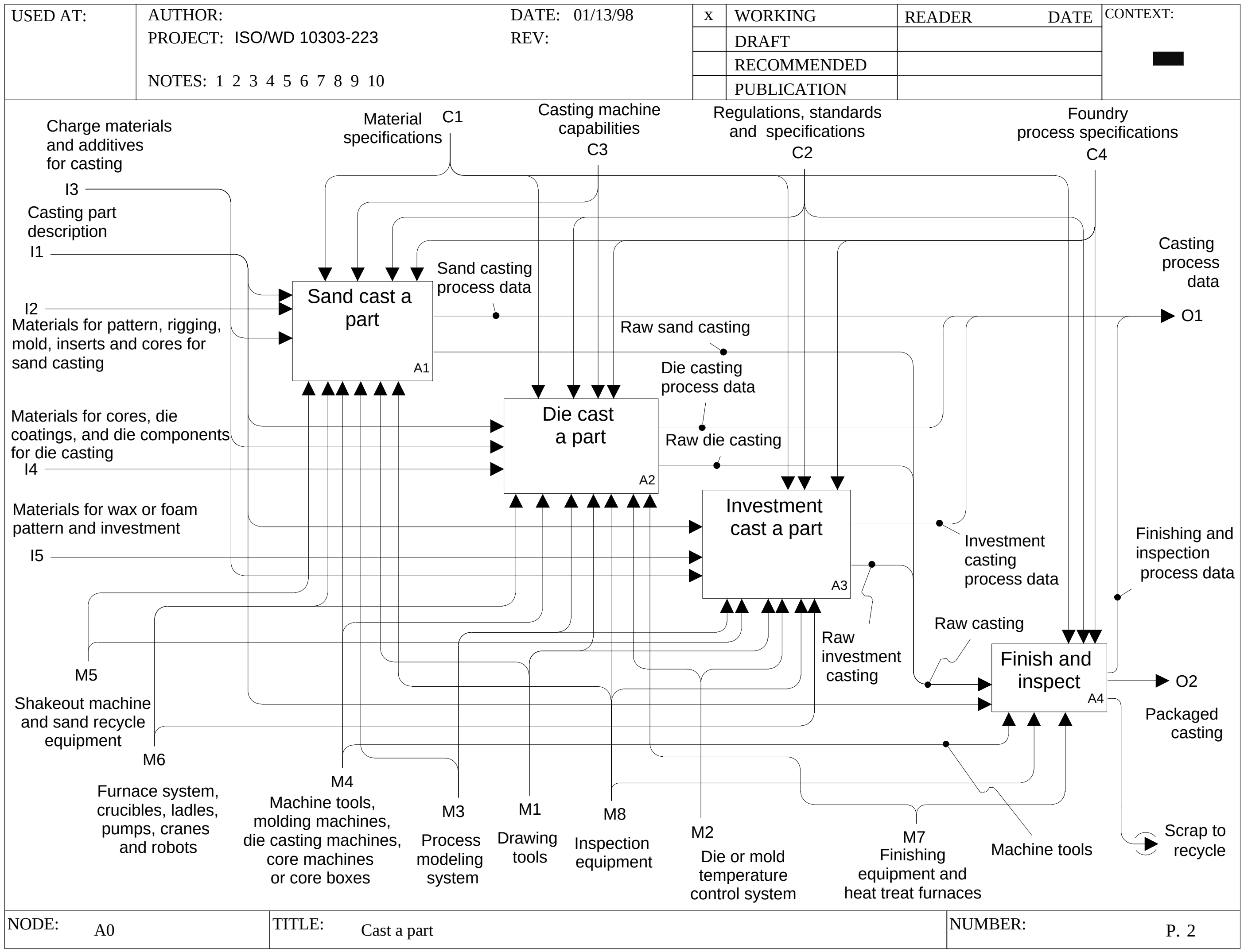


Figure F.3 – A0 Cast a Part

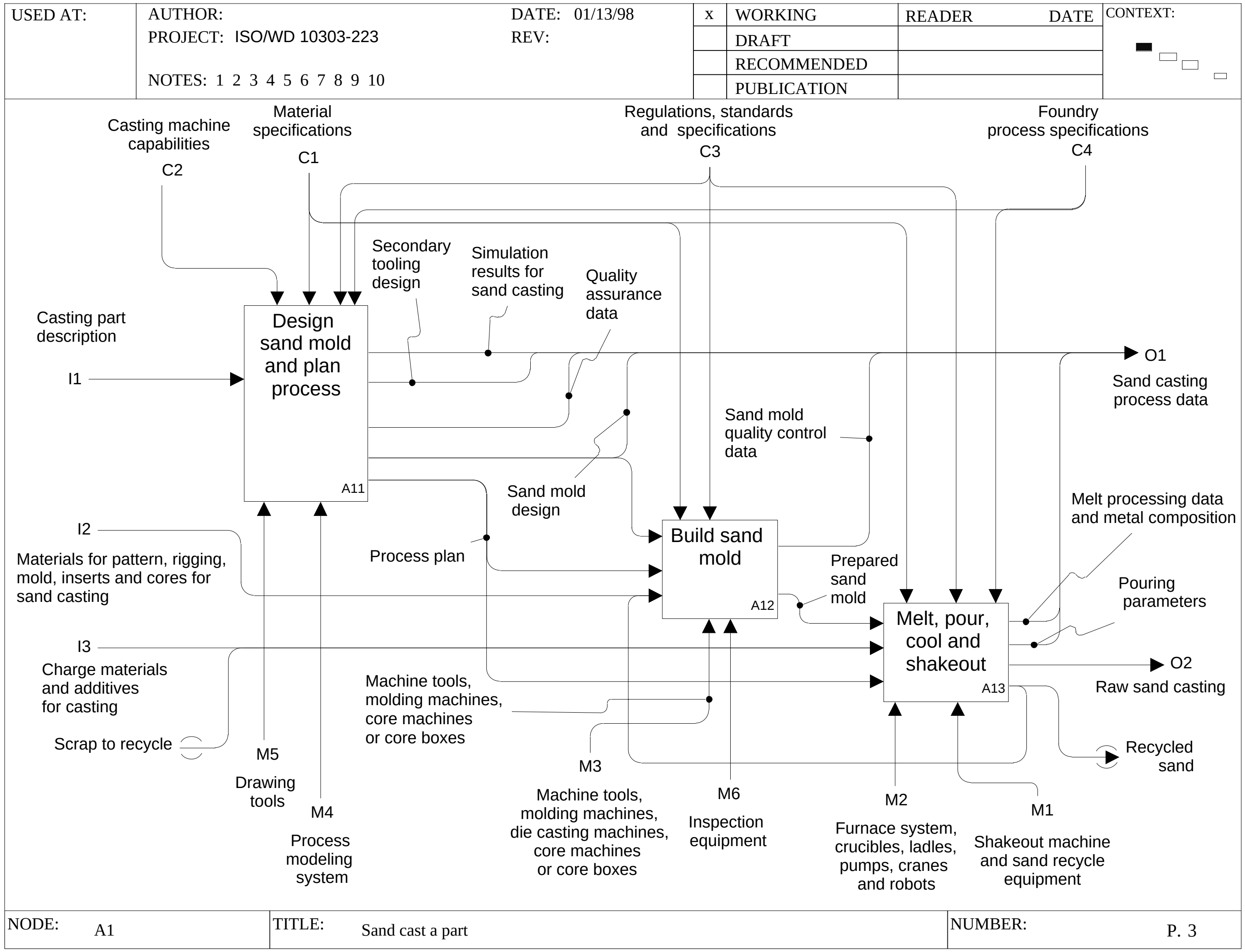


Figure F.4 – A1 Sand Cast a Part

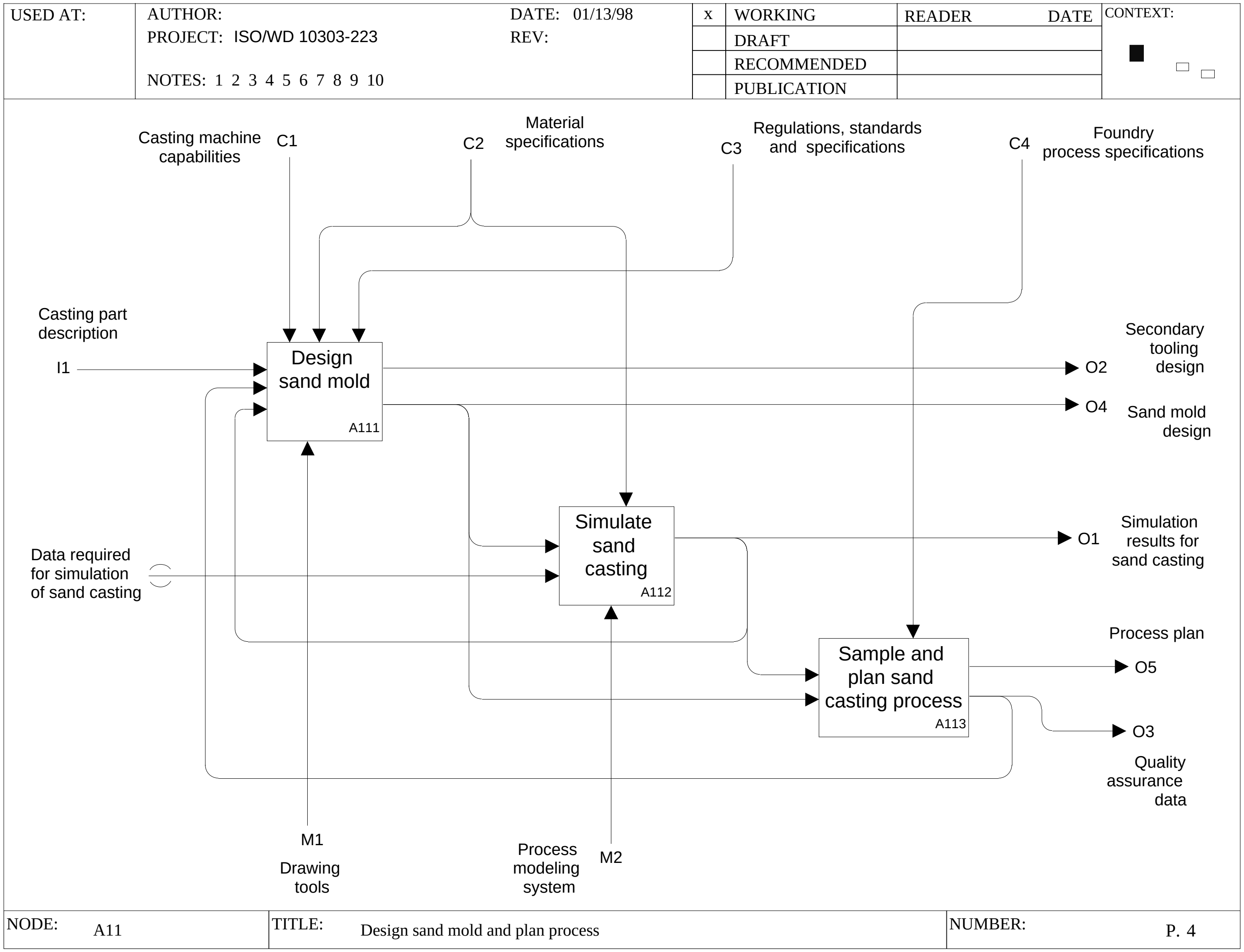


Figure F.5 – A11 Design Sand Mold and Plan Process

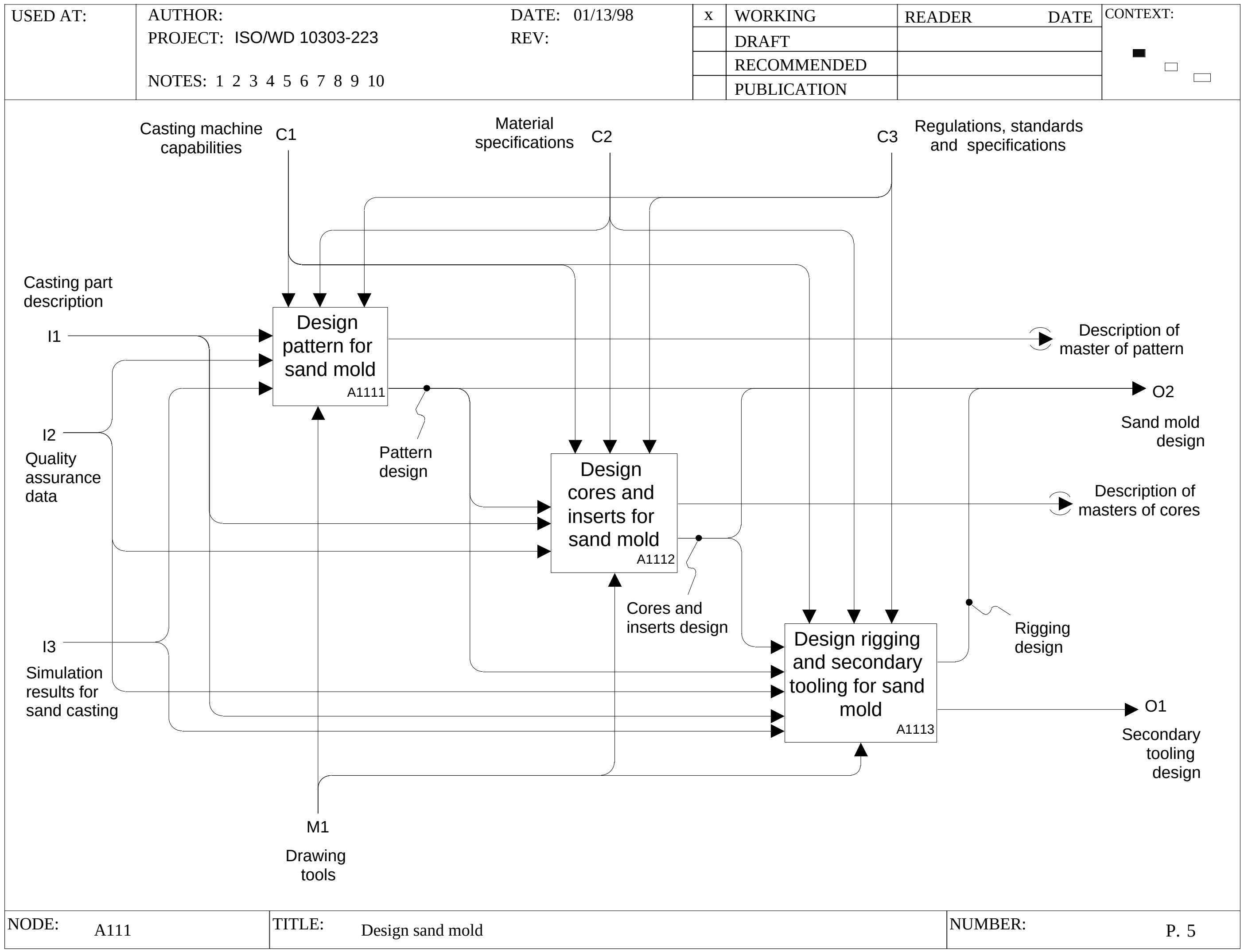


Figure F.6 – A111 Design Sand Mold

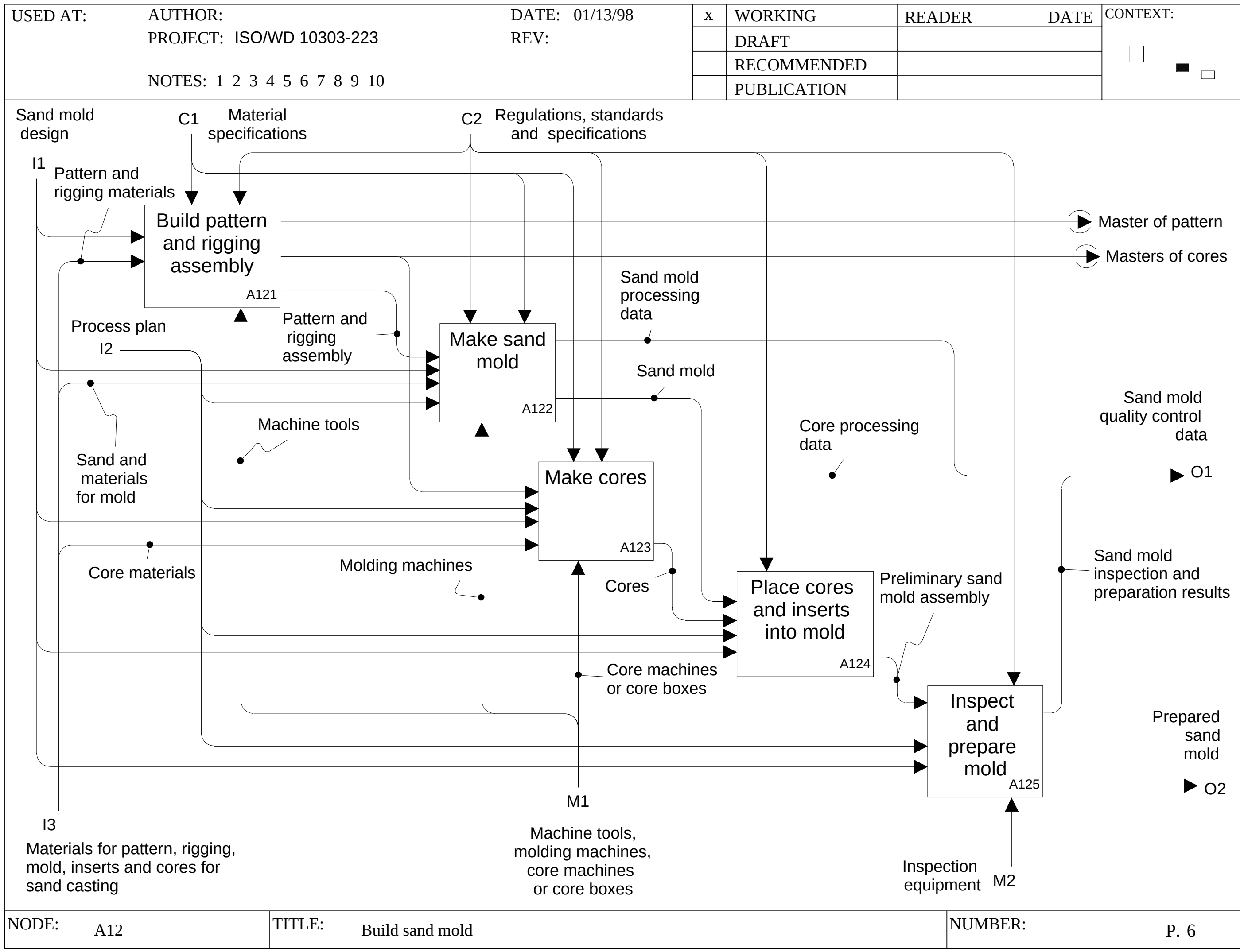


Figure F.7 – A12 Build Sand Mold

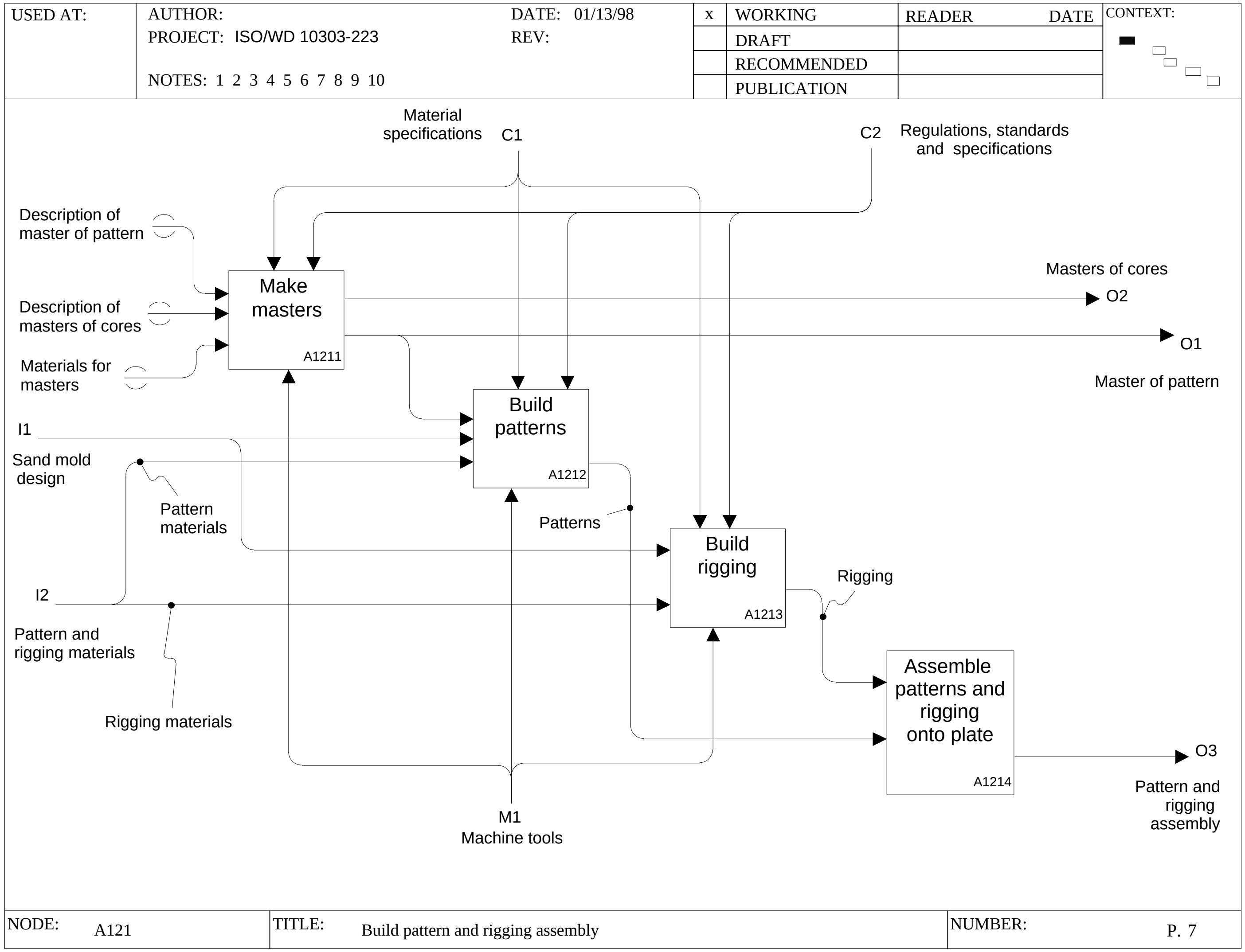


Figure F.8 – A121 Build Pattern and Rigging Assembly

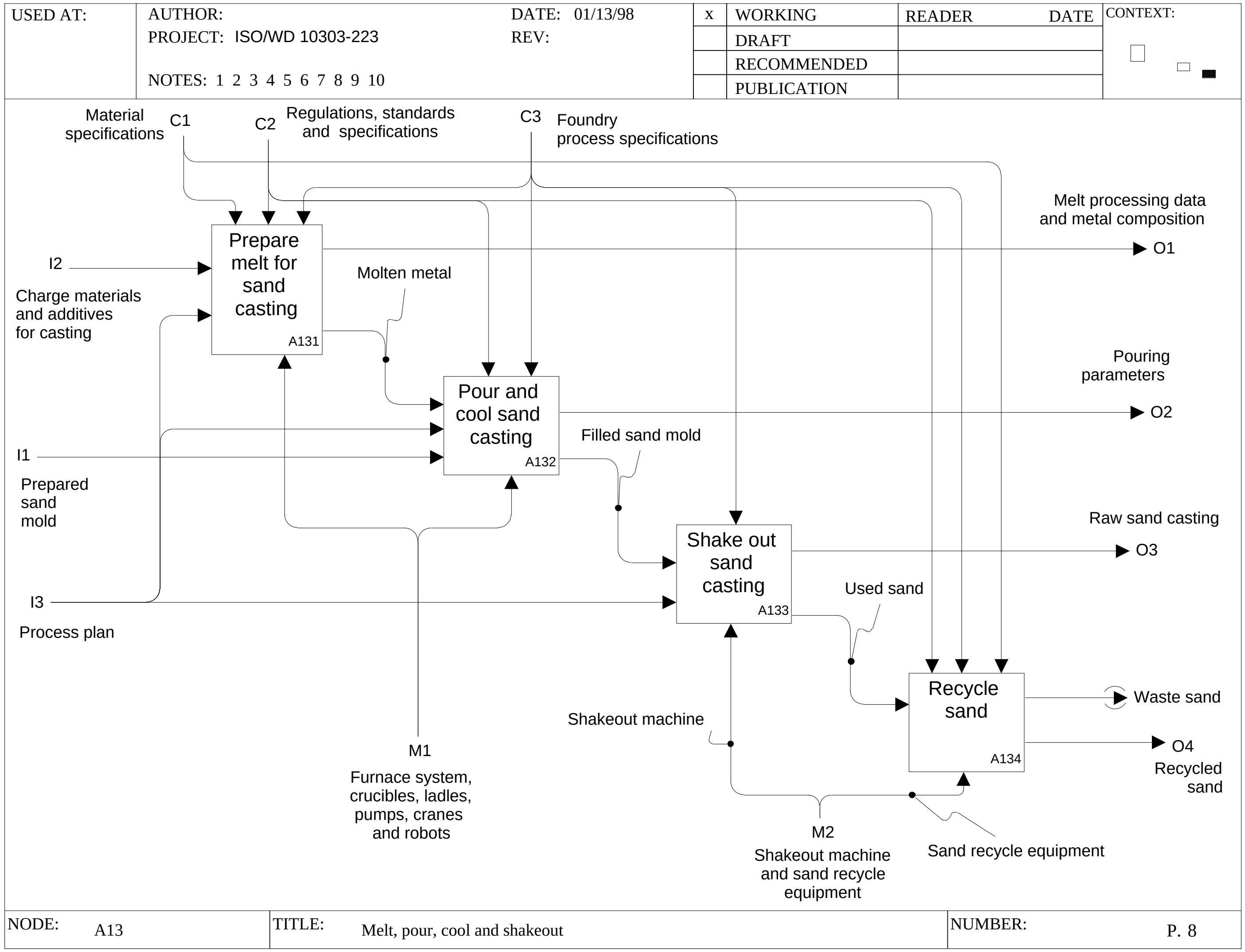
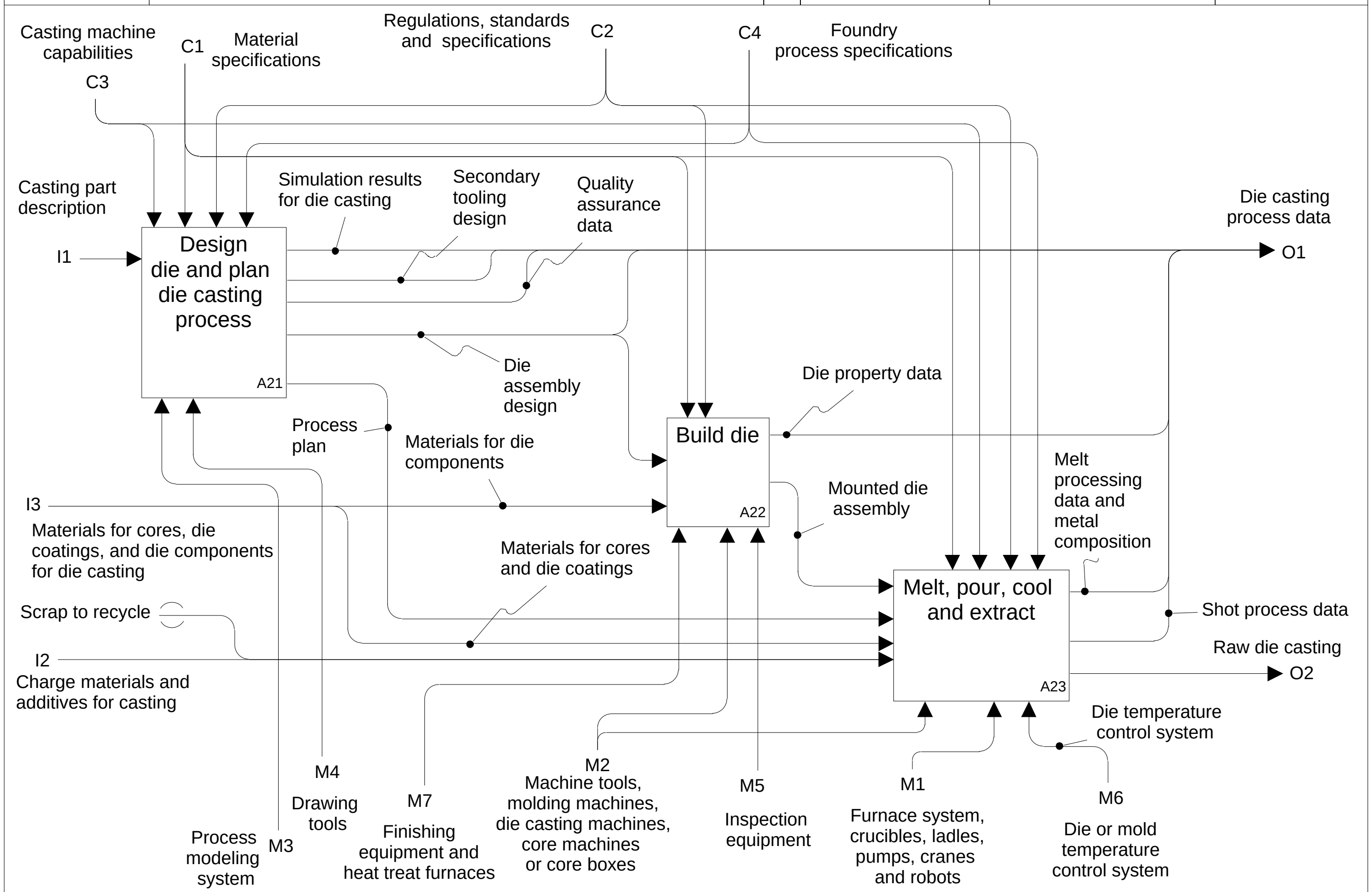


Figure F.9 – A13 Melt, Pour Cool and Shakeout

USED AT:	AUTHOR: PROJECT: ISO/WD 10303-223 NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 01/13/98 REV:	x	WORKING	READER	DATE	CONTEXT:
				DRAFT			
				RECOMMENDED			
				PUBLICATION			



NODE: A2	TITLE: Die cast a part	NUMBER: P. 9

Figure F.10 – A2 Die Cast a Part

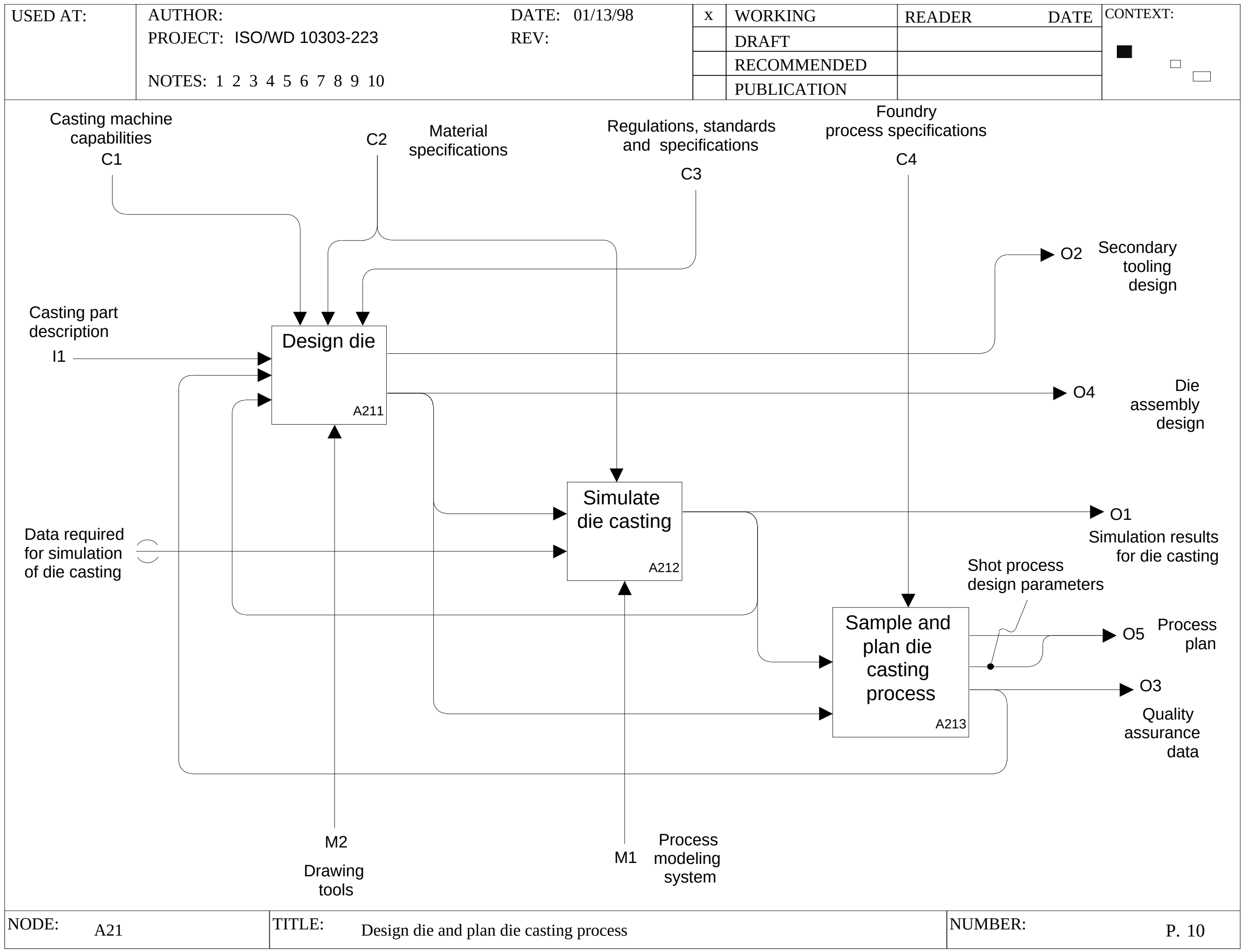


Figure F.11 – A21 Design Die and Plan Die Casting Process

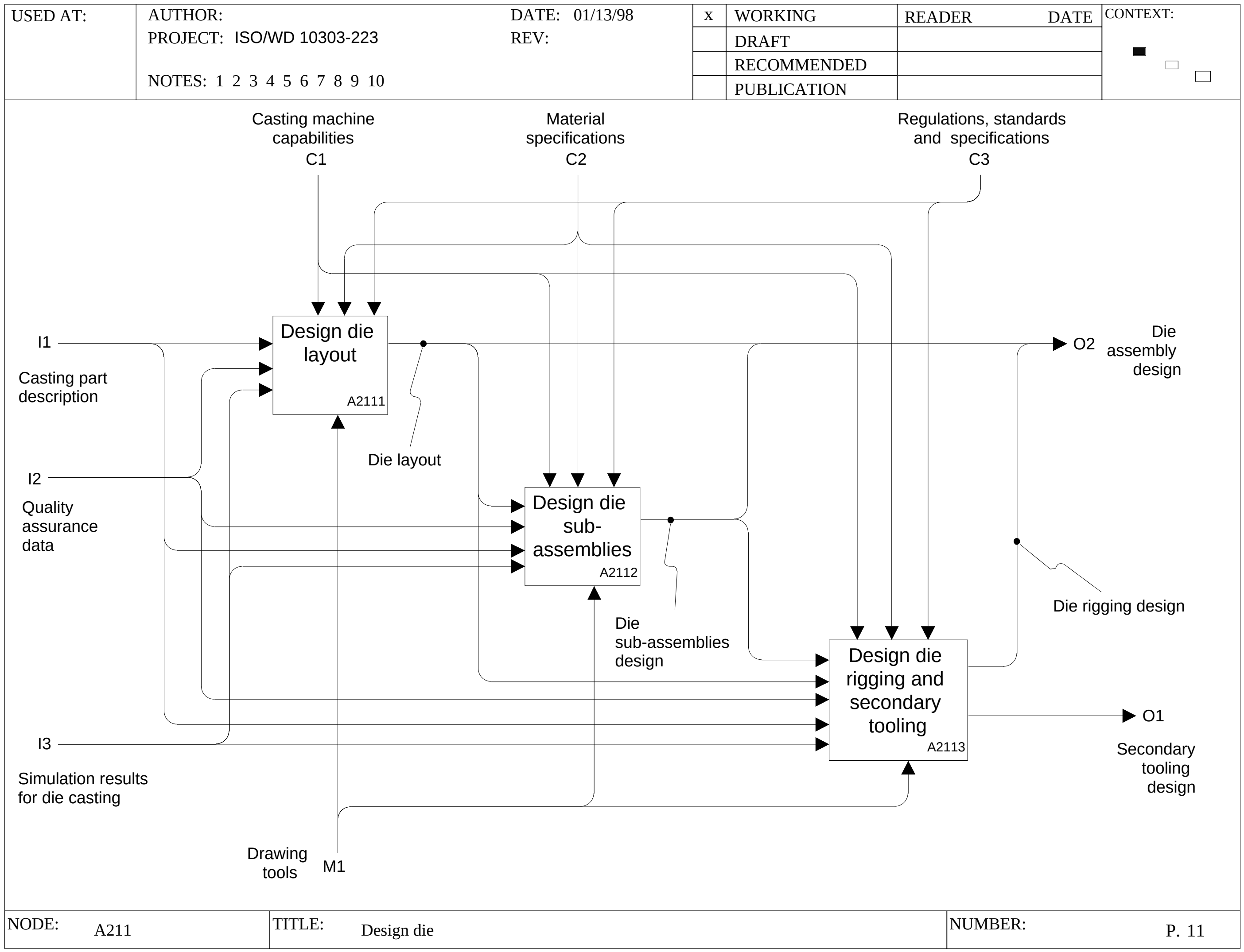


Figure F.12 – A211 Design Die

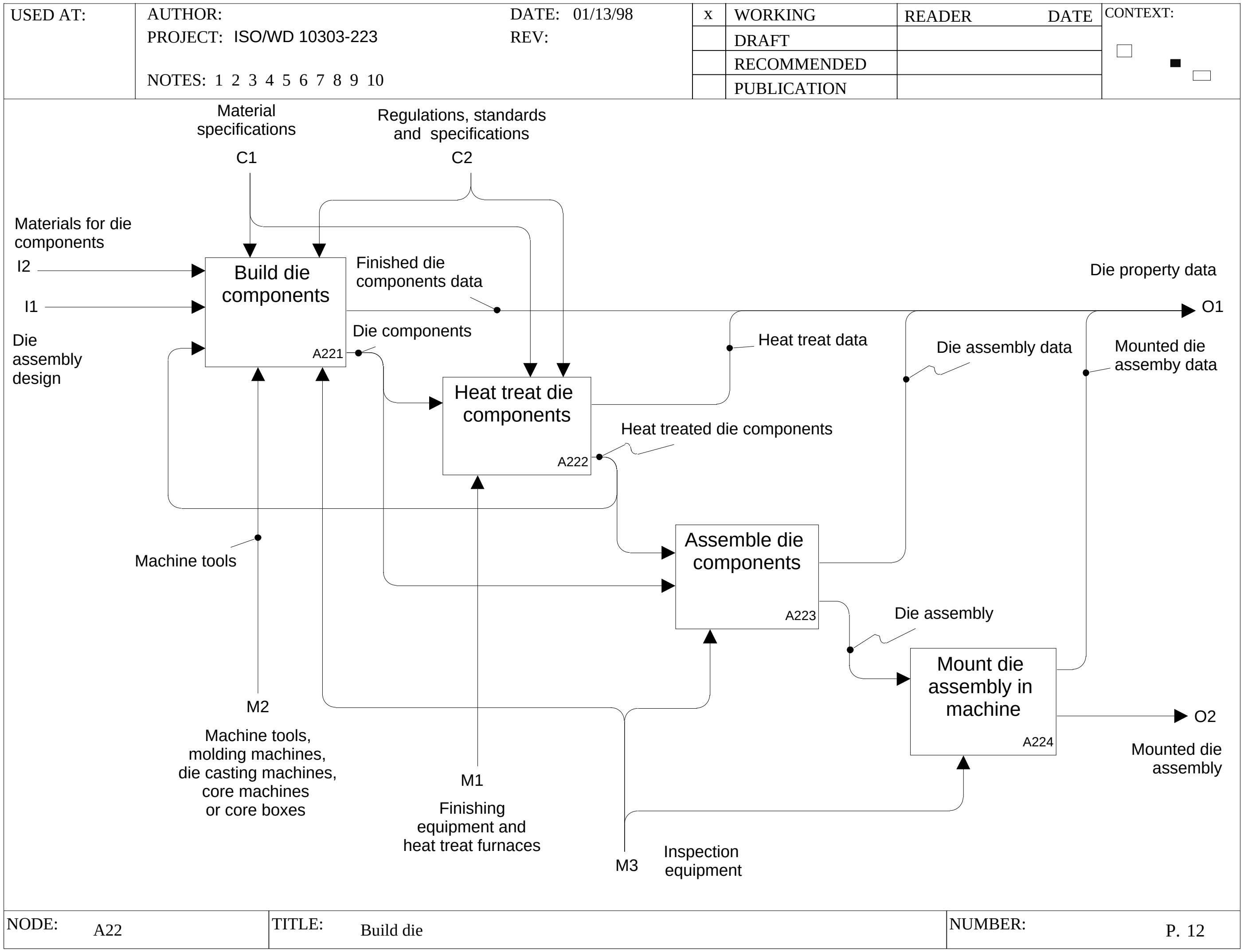


Figure F.13 – A22 Build Die

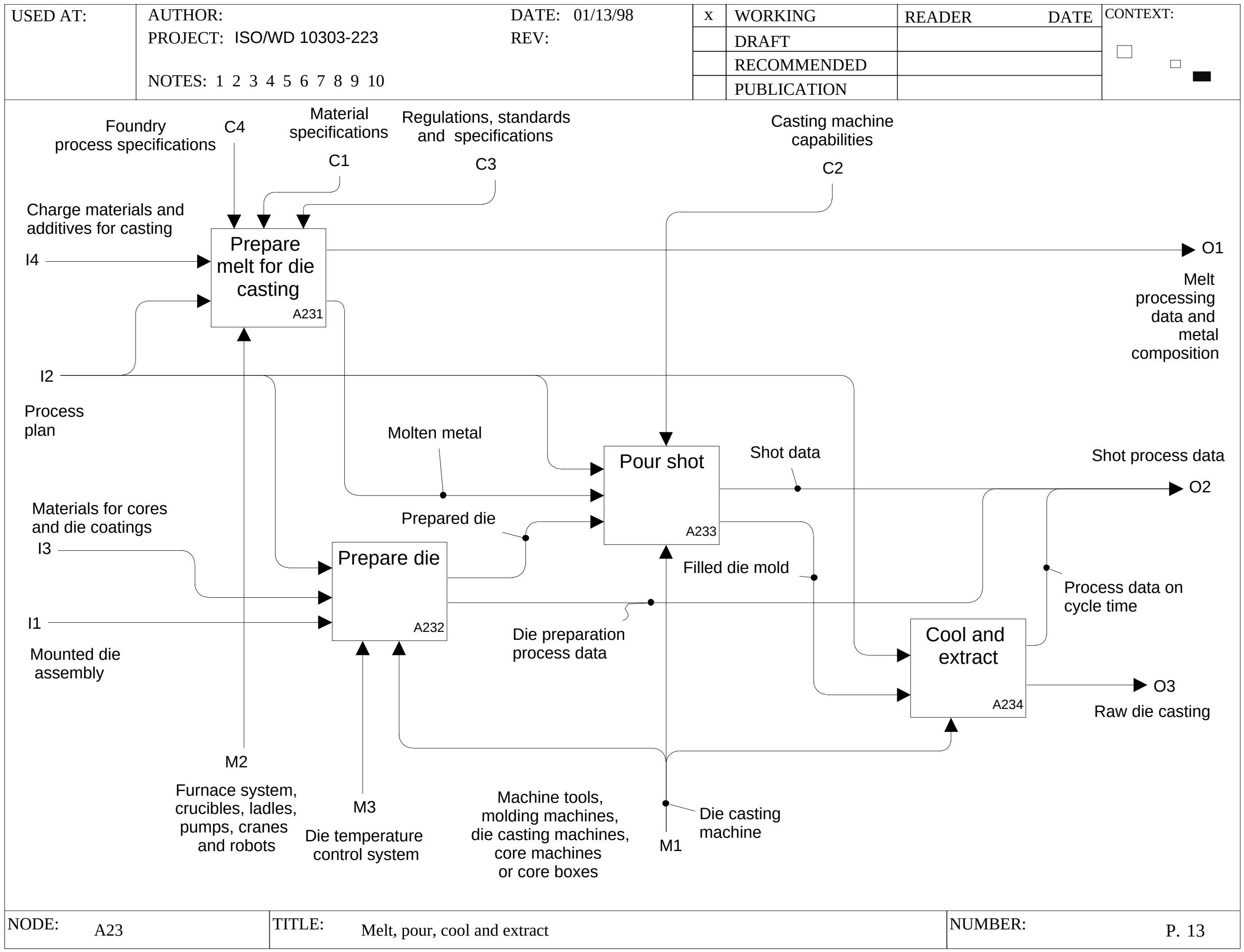
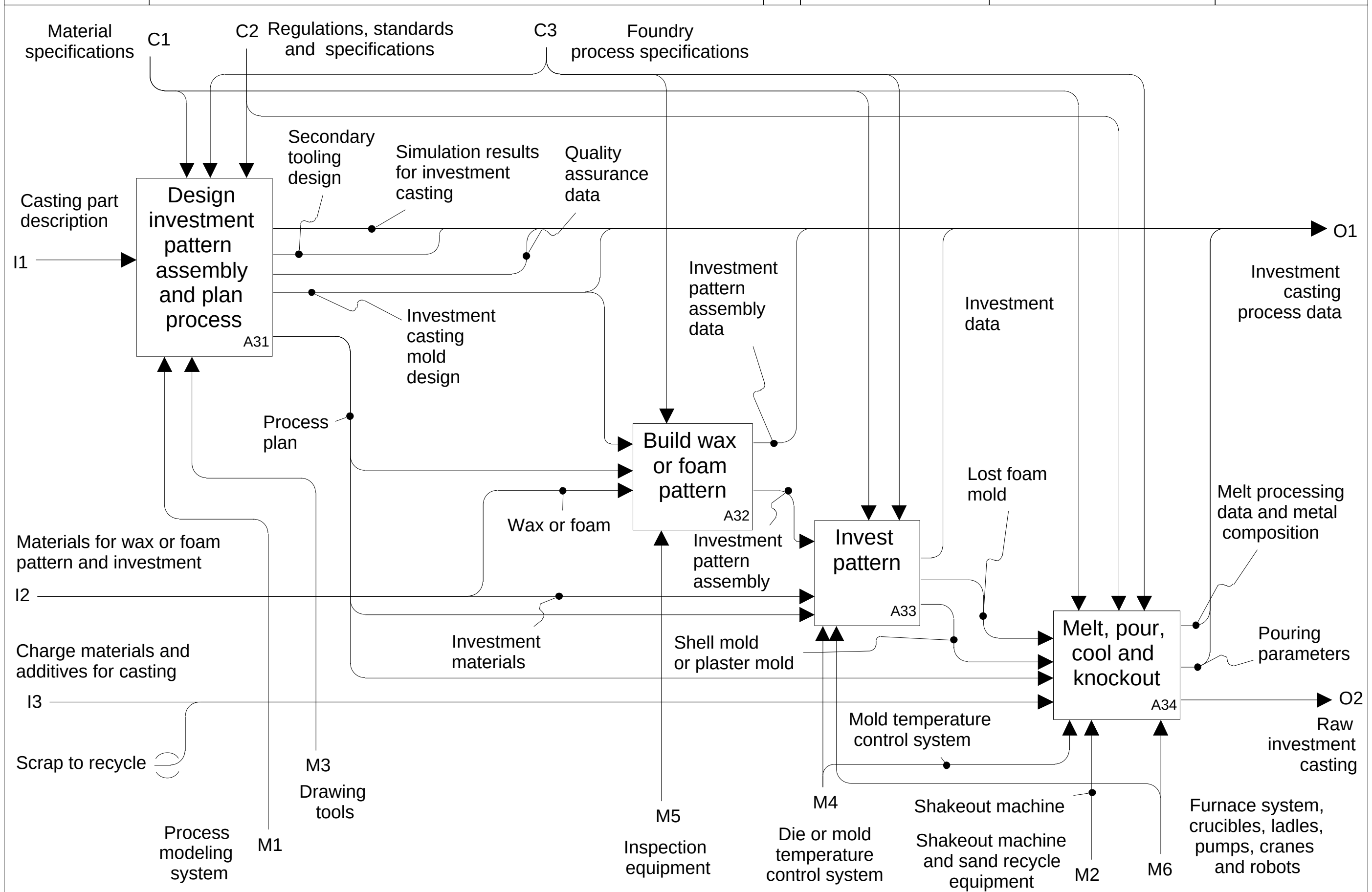


Figure F.14 – A23 Melt, Pour, Cool and Extract

USED AT:	AUTHOR: PROJECT: ISO/WD 10303-223 NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 01/13/98 REV:	x WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE	CONTEXT:



NODE: A3	TITLE: Investment cast a part	NUMBER: P. 14

Figure F.15 – A3 Investment Cast a Part

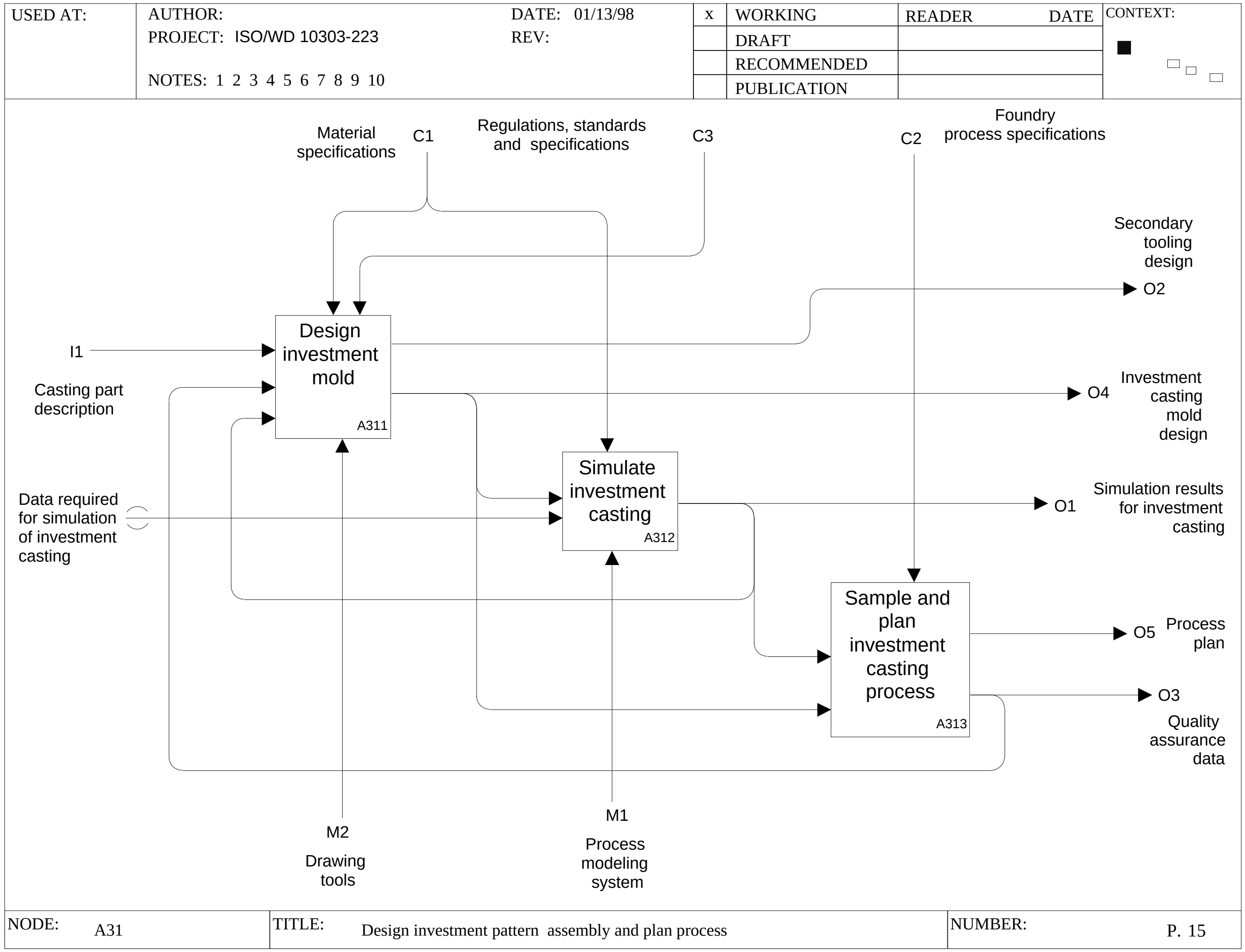


Figure F.16 – A31 Design Investment Pattern Assembly and Plan Process

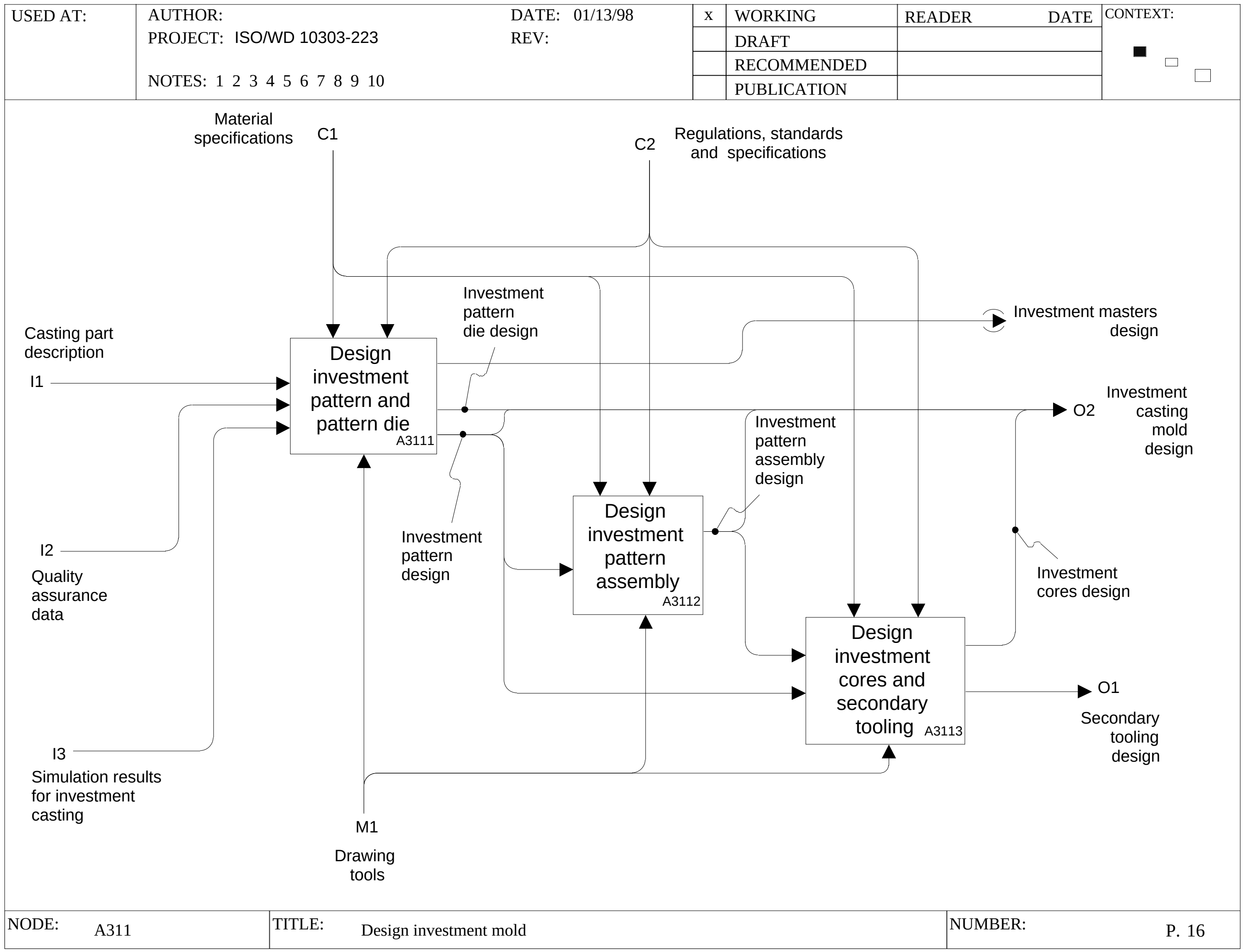


Figure F.17 – A311 Design Investment Mold

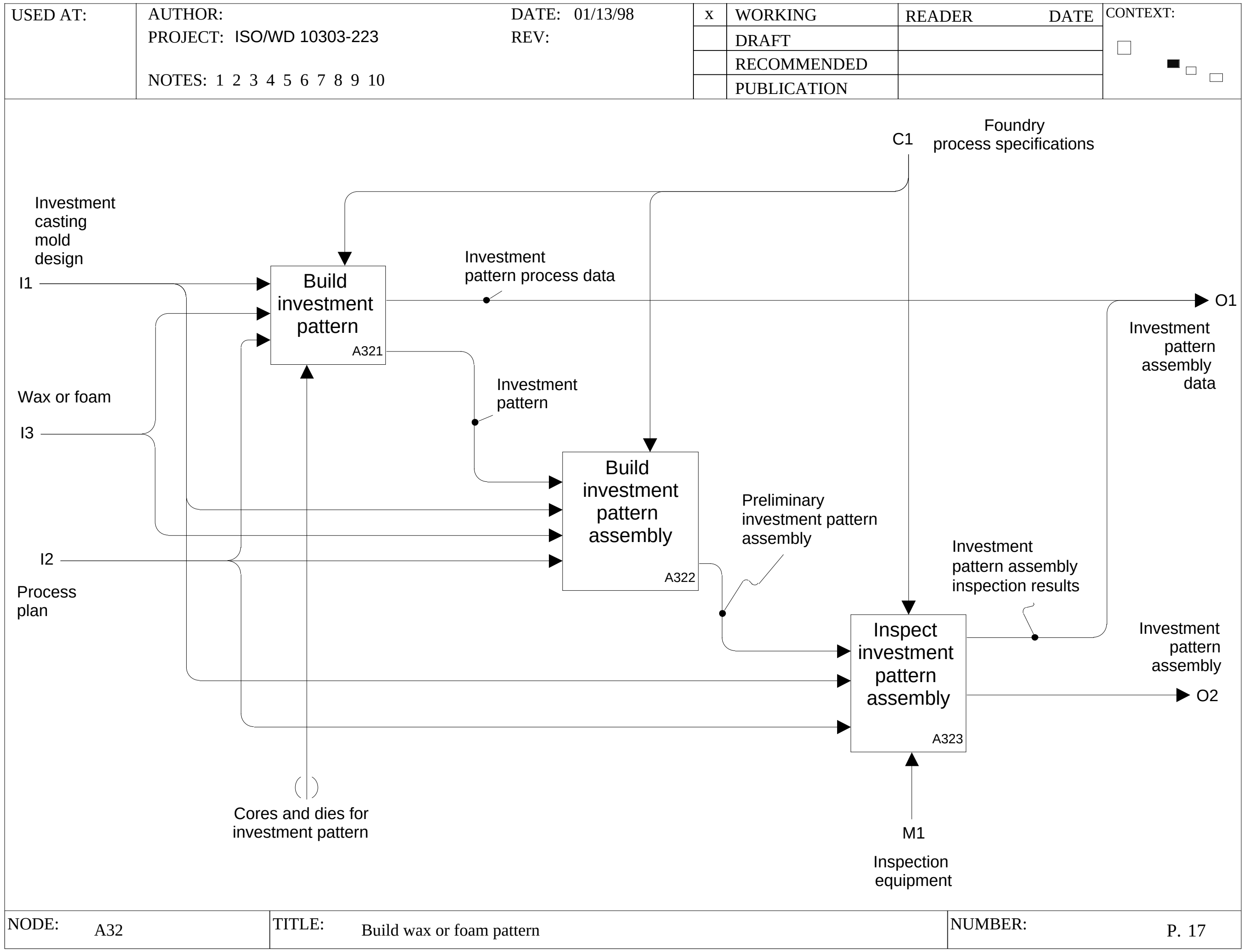


Figure F.18 – A32 Build Wax or Foam Pattern

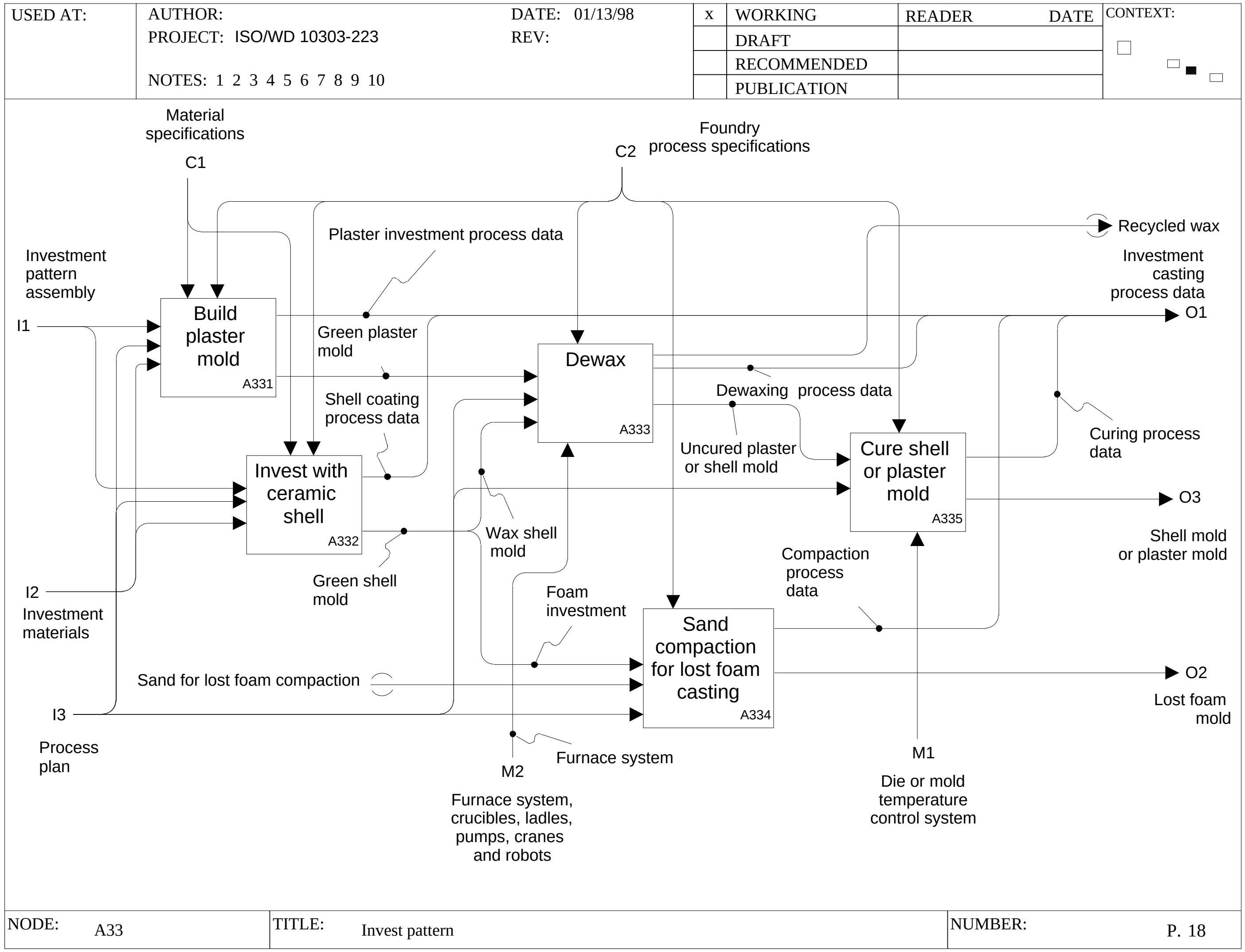


Figure F.19 – A33 Invest Pattern

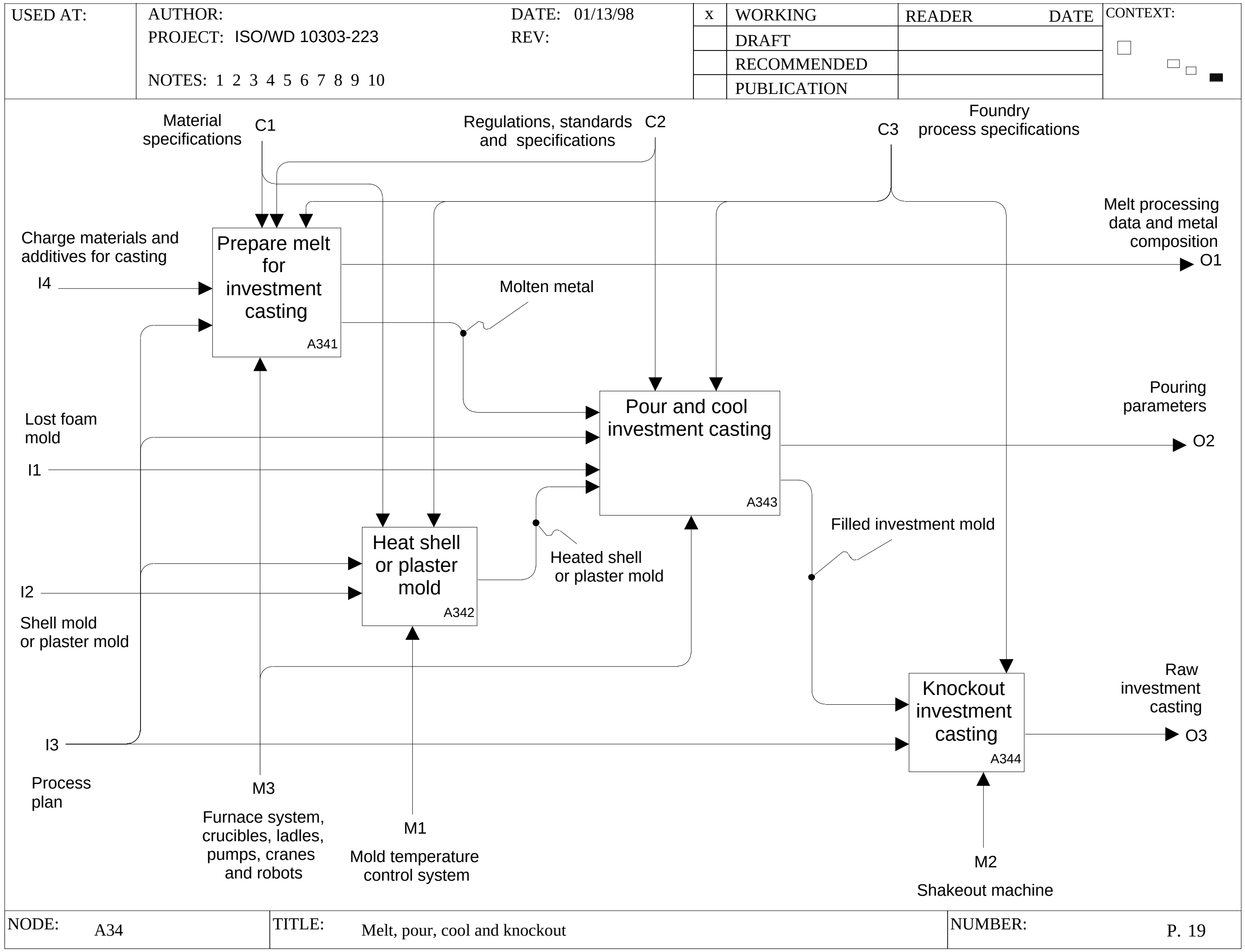


Figure F.20 – A34 Melt, Pour, Cool and Knockout

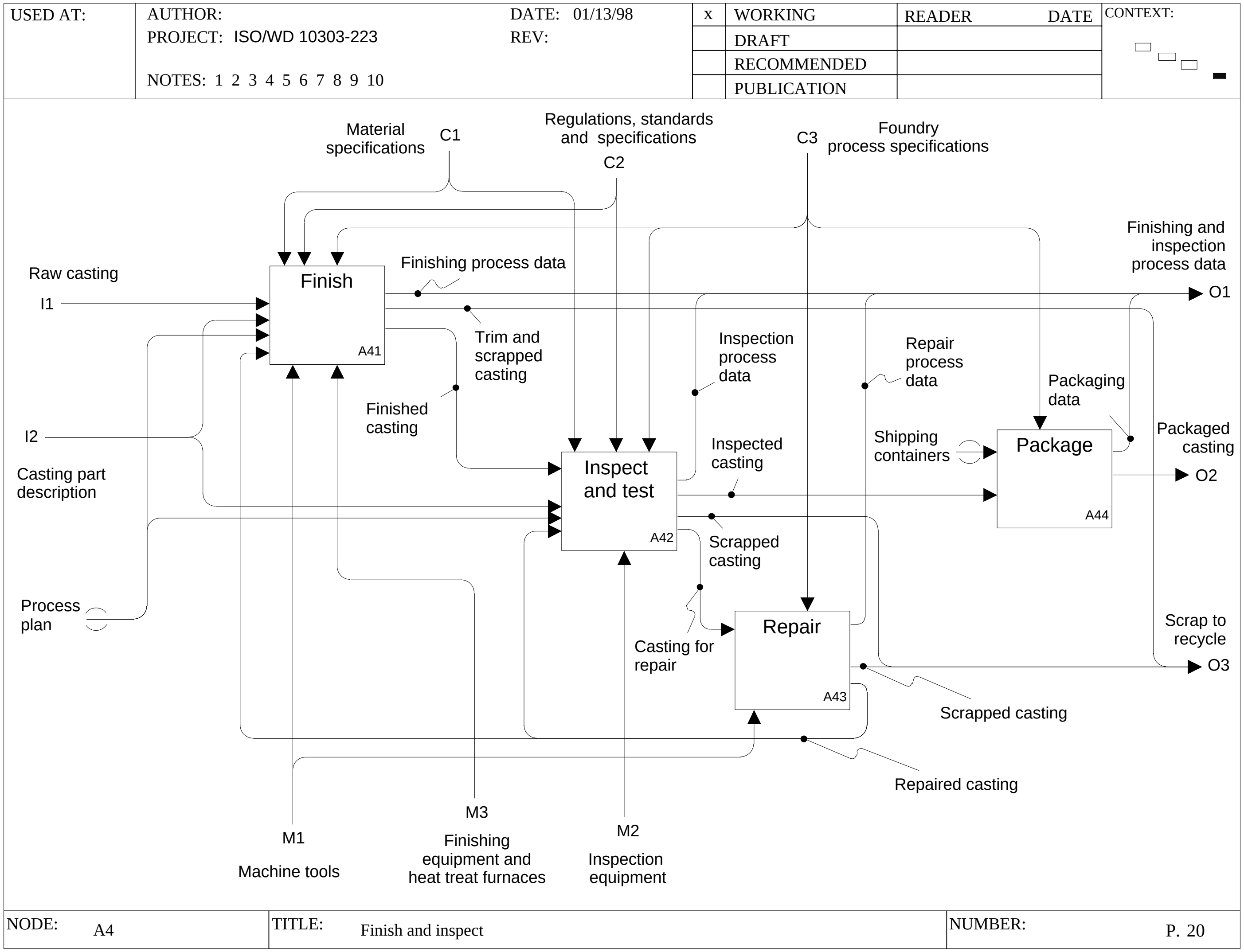


Figure F.21 – A4 Finish and Inspect

Annex G
(informative)

Application reference model

This annex provides the application reference model for this Part of ISO 10303 and is given in figures G.1 through G.19. The application reference model is a graphical representation of the structure and constraints of the application objects specified in clause 4.2. The graphical form of the application reference model is presented in EXPRESS-G. The application reference model is independent from any implementation method.

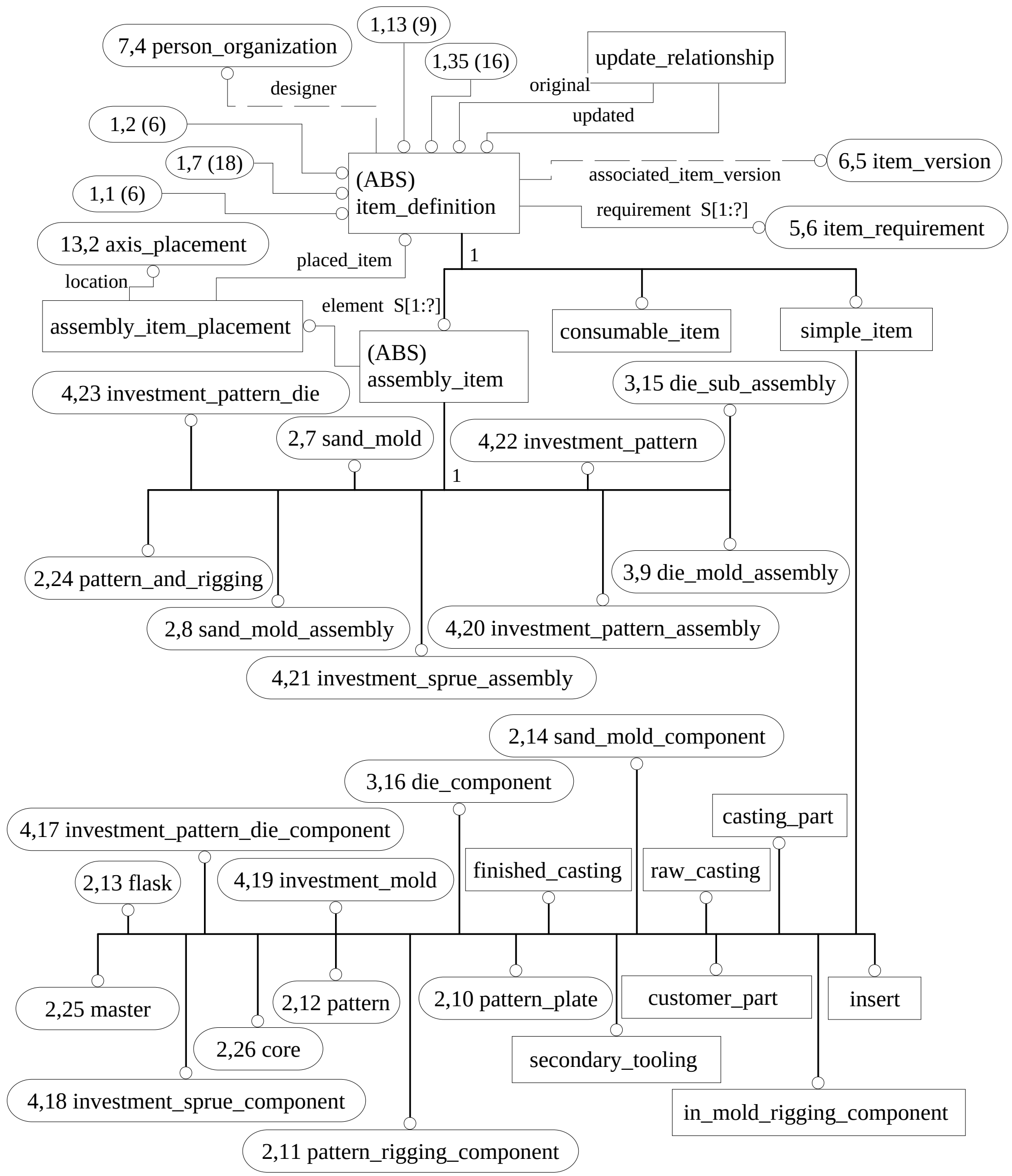


Figure G.1 – General Item UoF

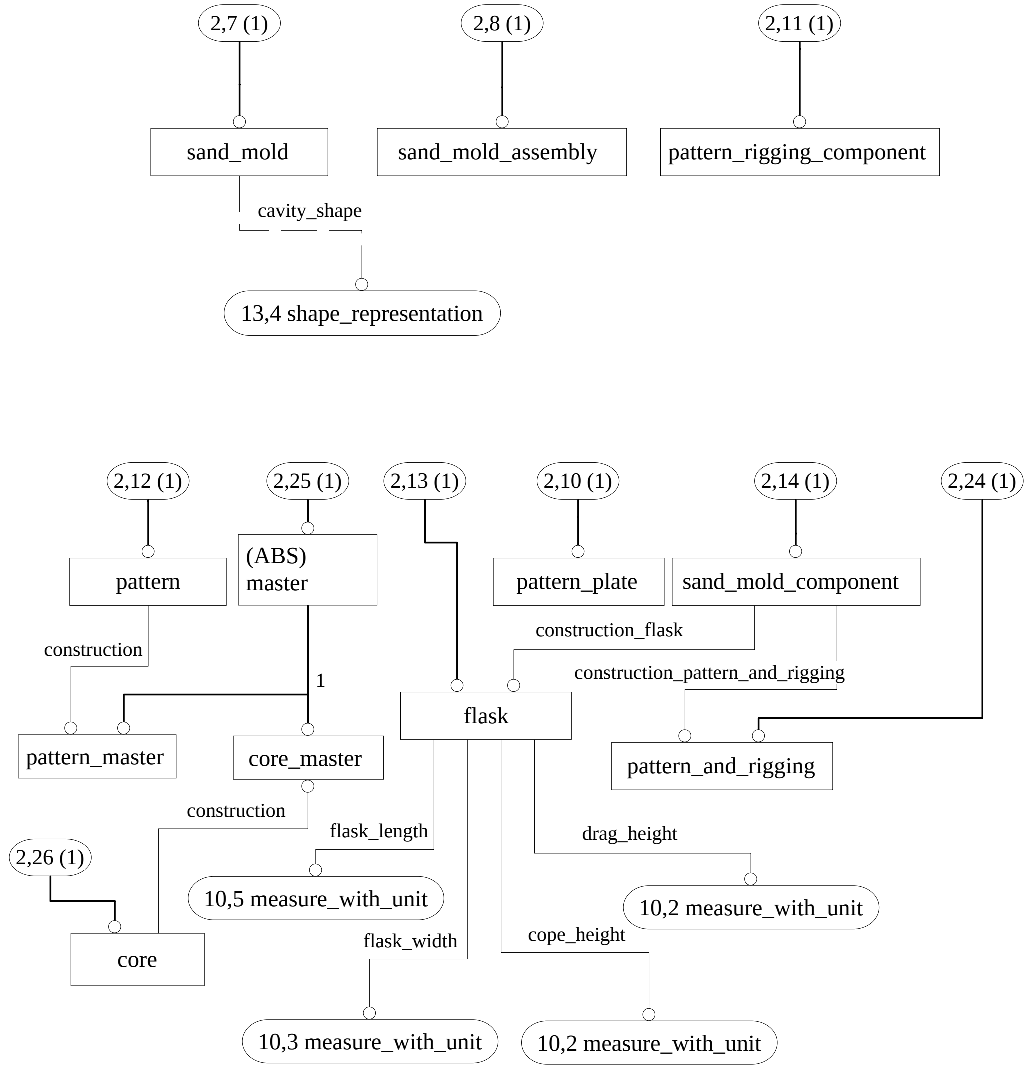


Figure G.2 – Sand Casting UoF

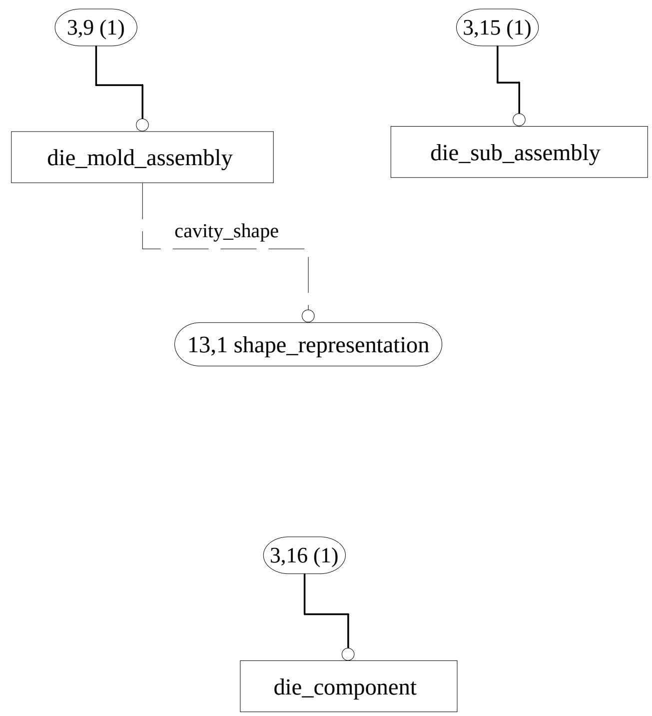


Figure G.3 – Die Casting UoF

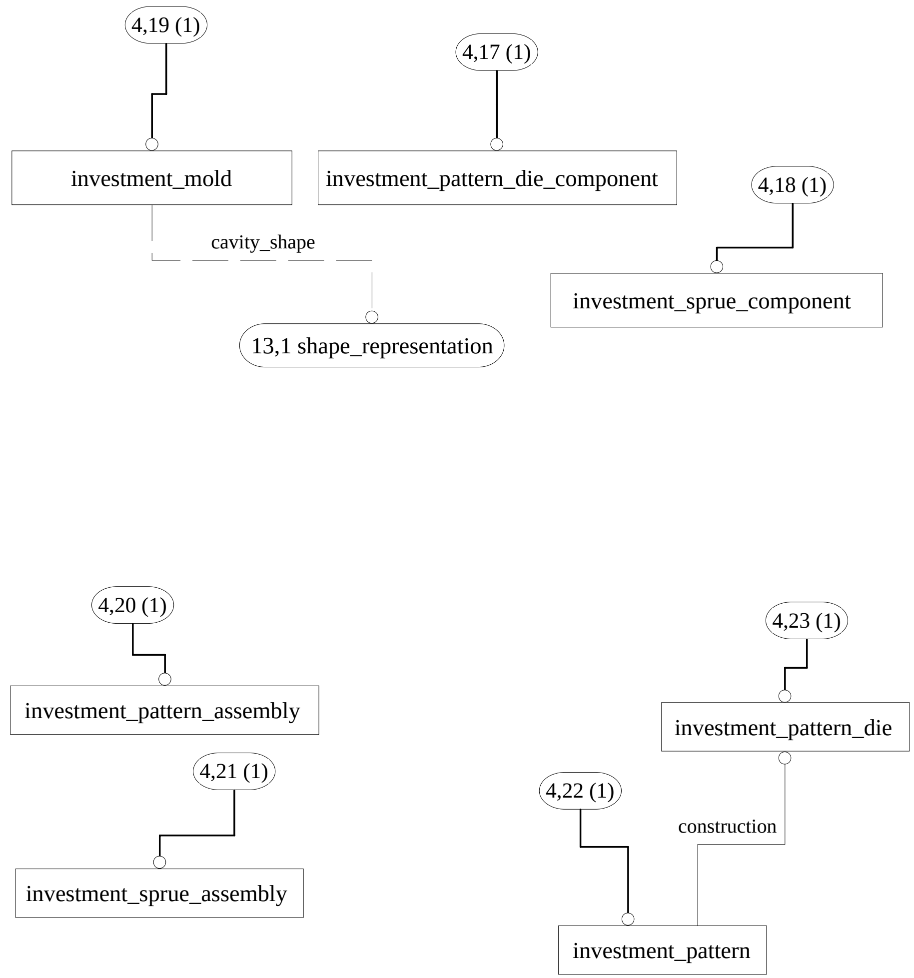


Figure G.4 – Investment Casting UoF

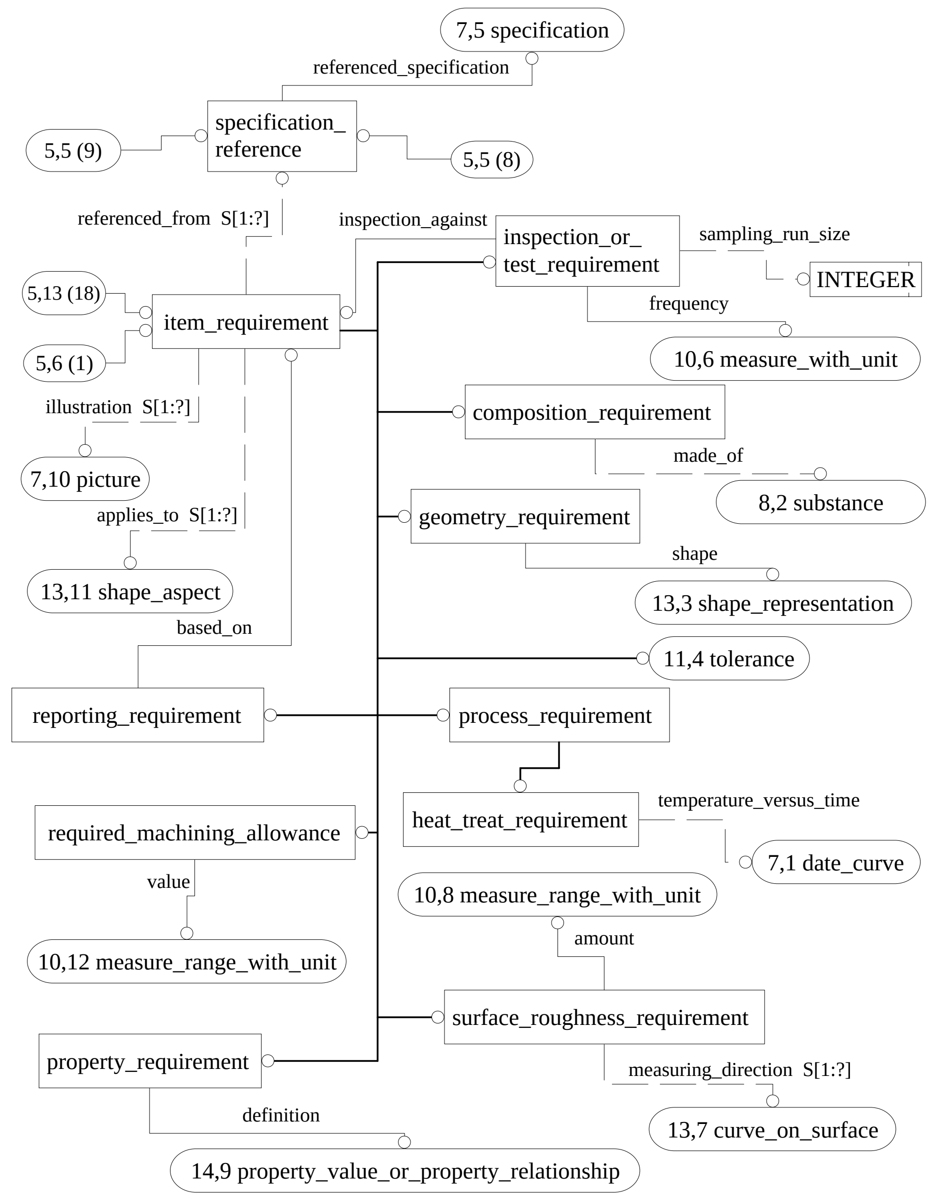


Figure G.5 – Requirement UoF

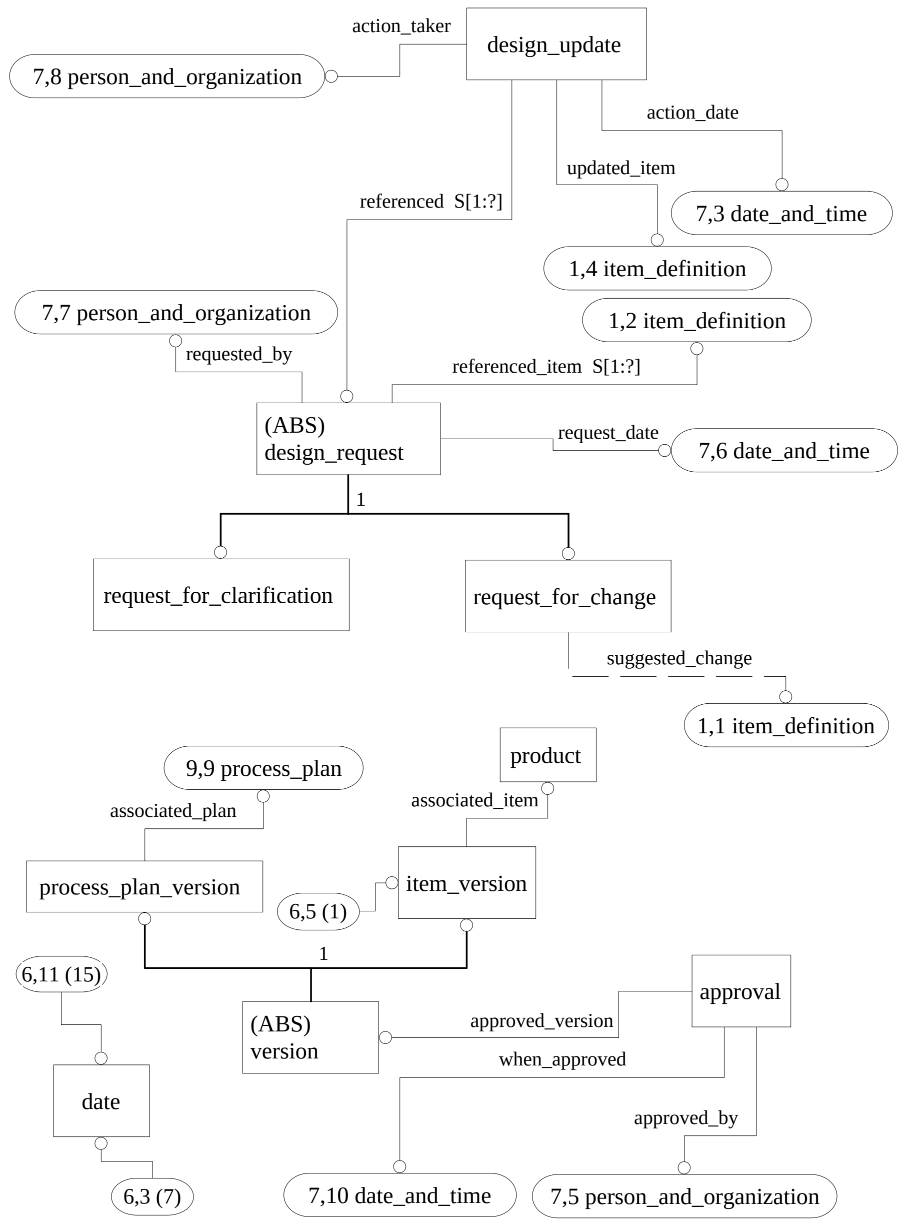


Figure G.6 – Configuration Management UoF

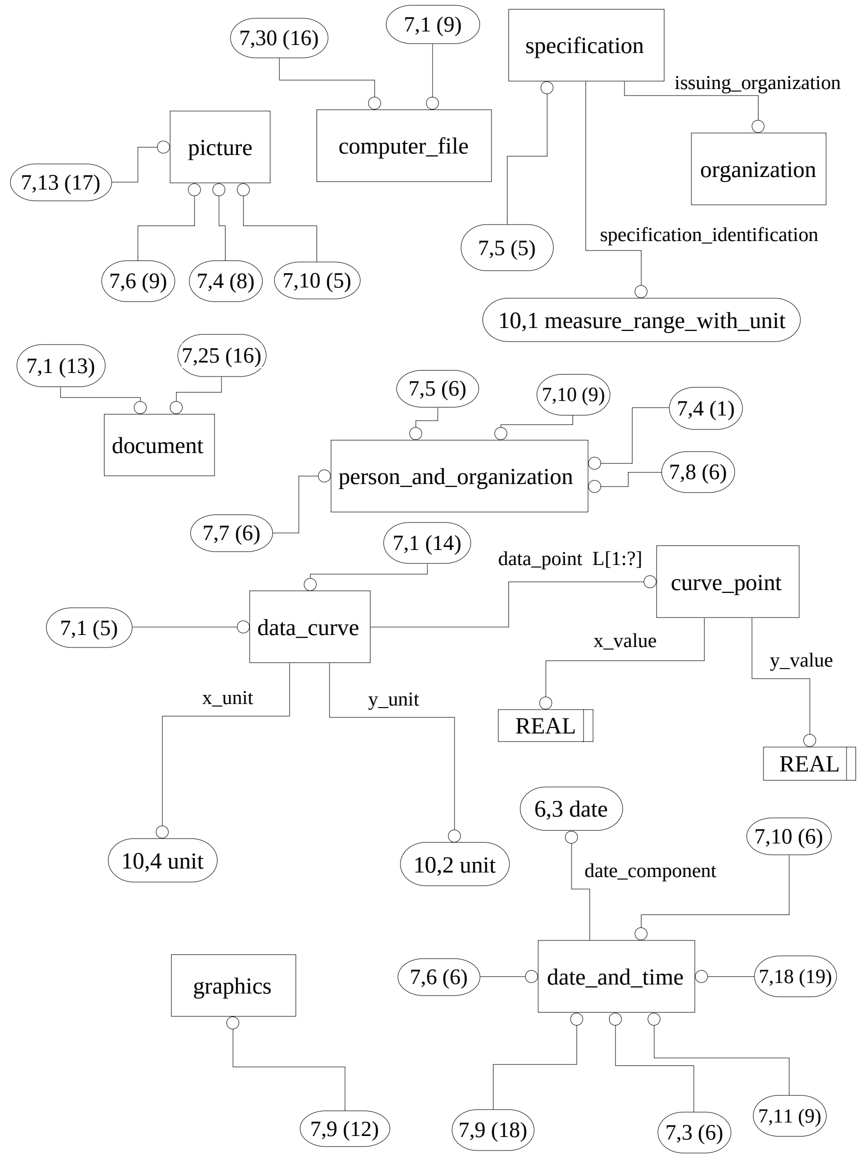


Figure G.7 – Basic Data UoF

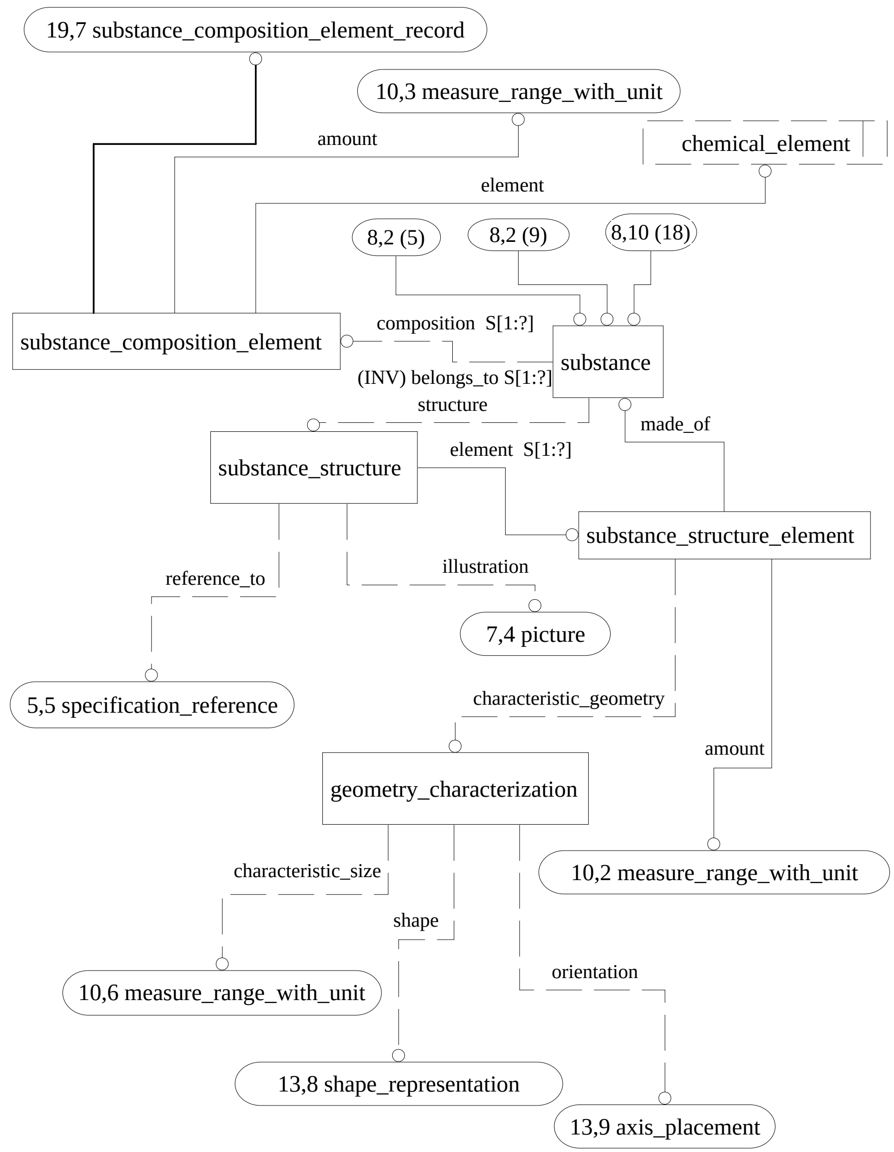


Figure G.8 – Substance Composition UoF

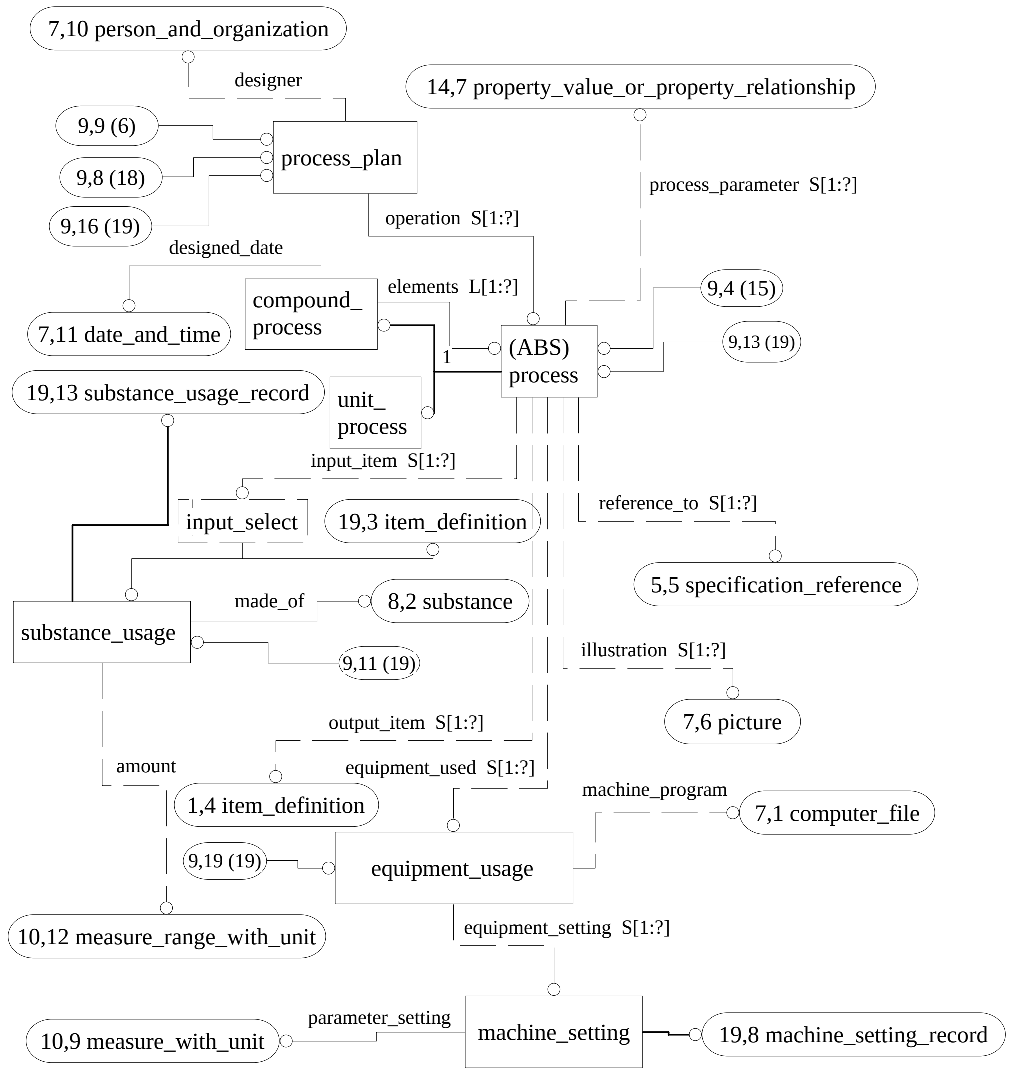


Figure G.9 – Process Plan UoF

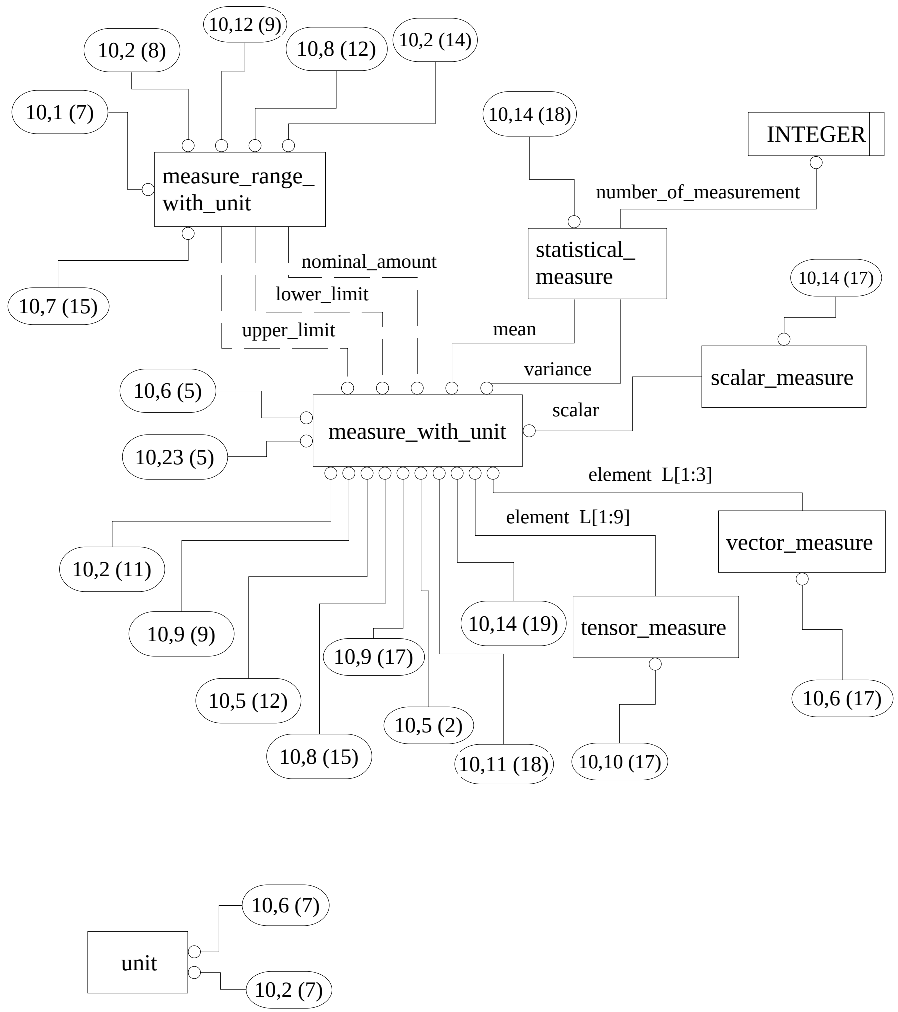


Figure G.10 – Measurement UoF

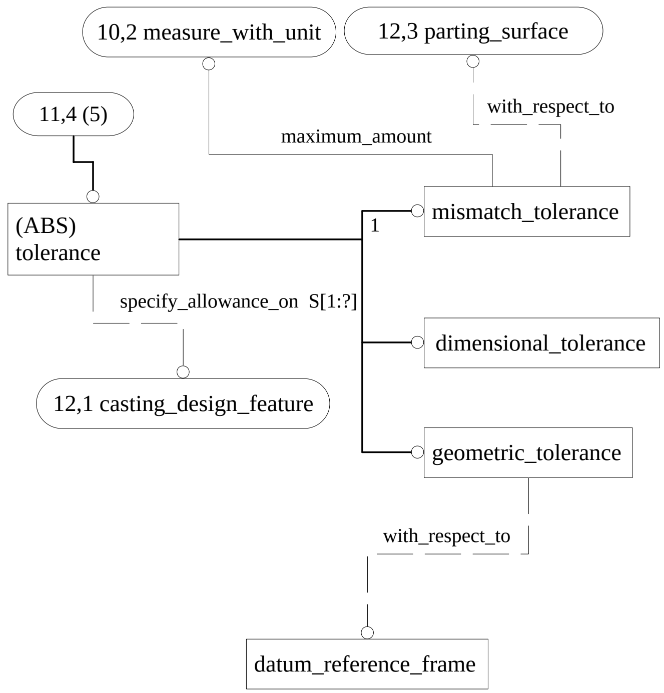


Figure G.11 – Tolerance UoF

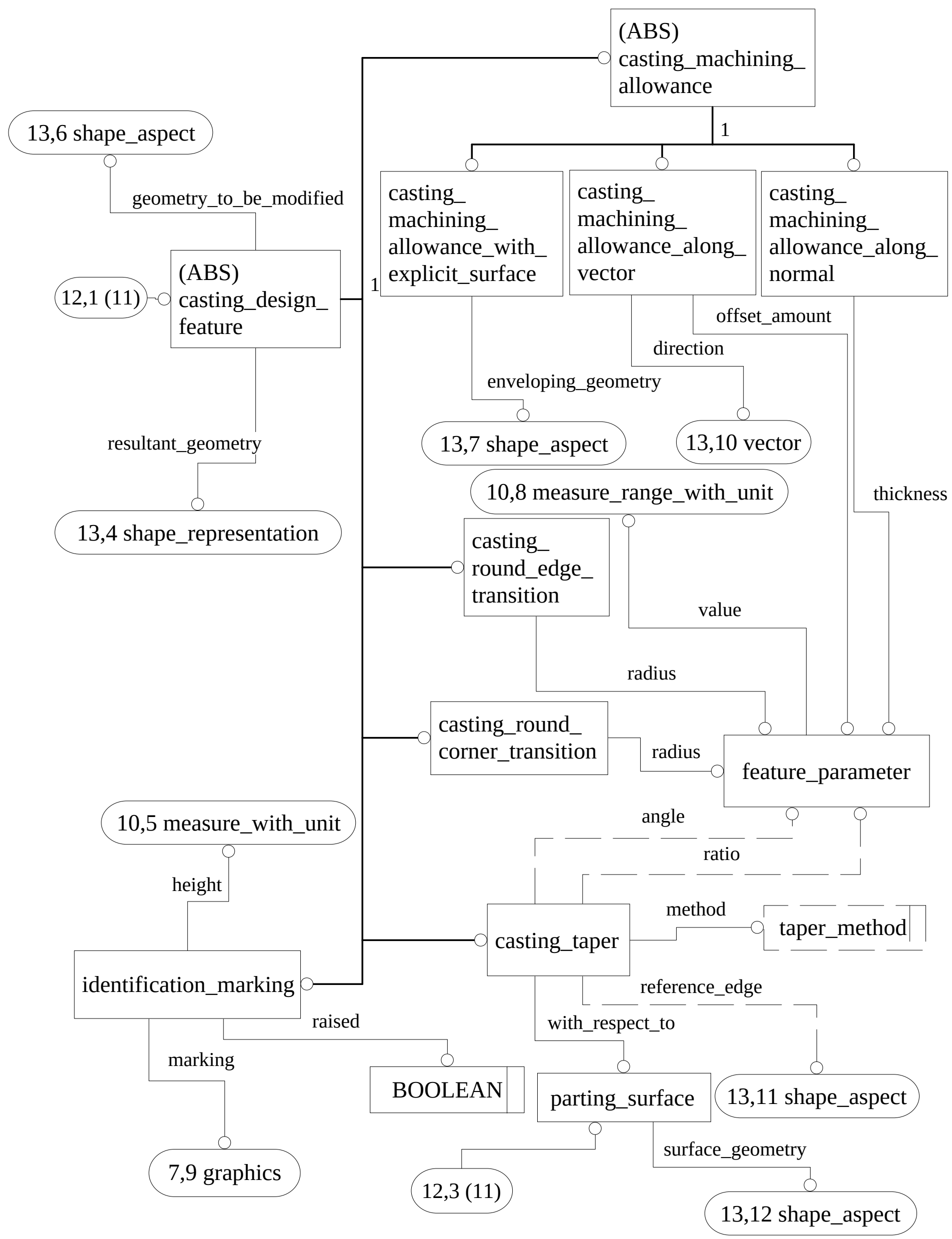


Figure G.12 – Feature UoF

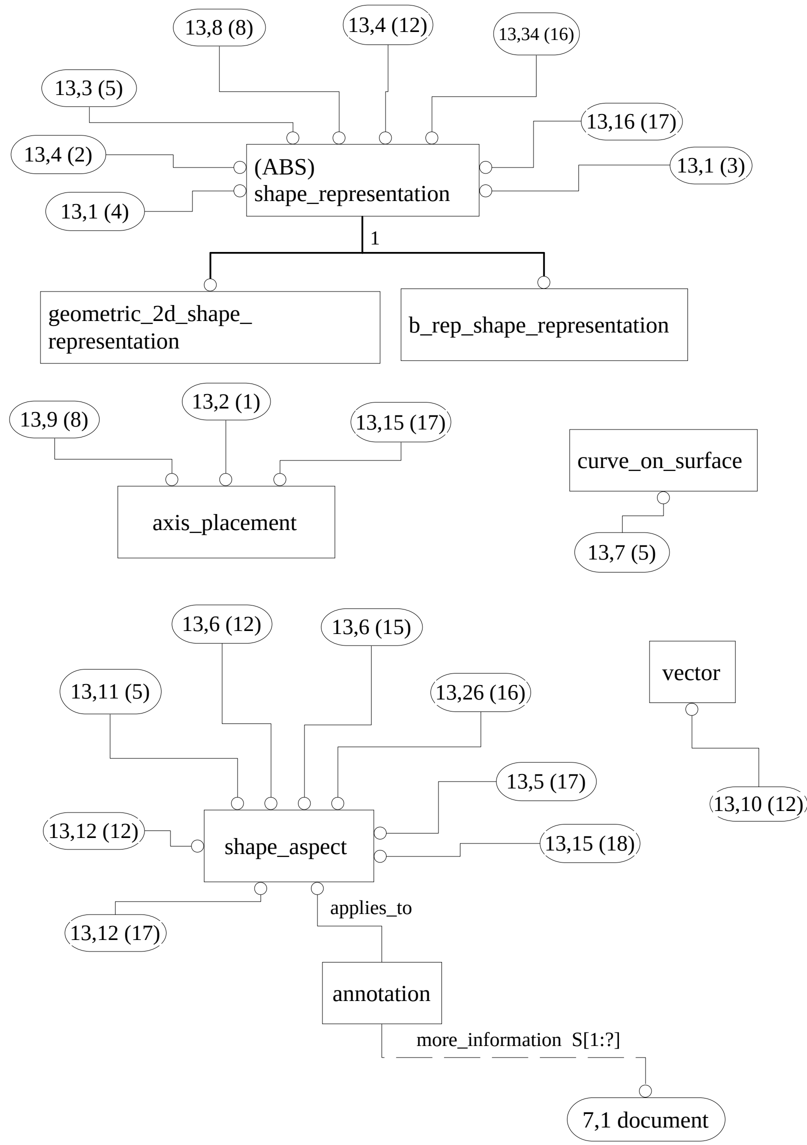


Figure G.13 – Geometry UoF

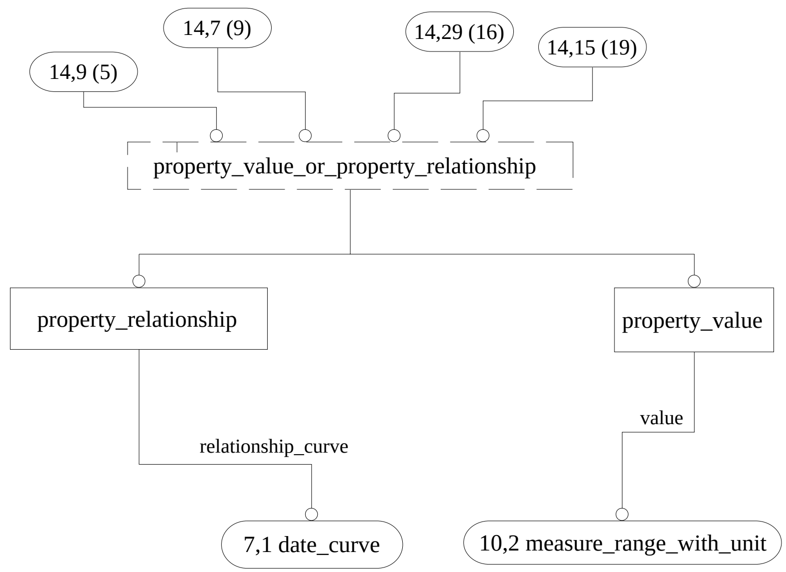


Figure G.14 – Property UoF

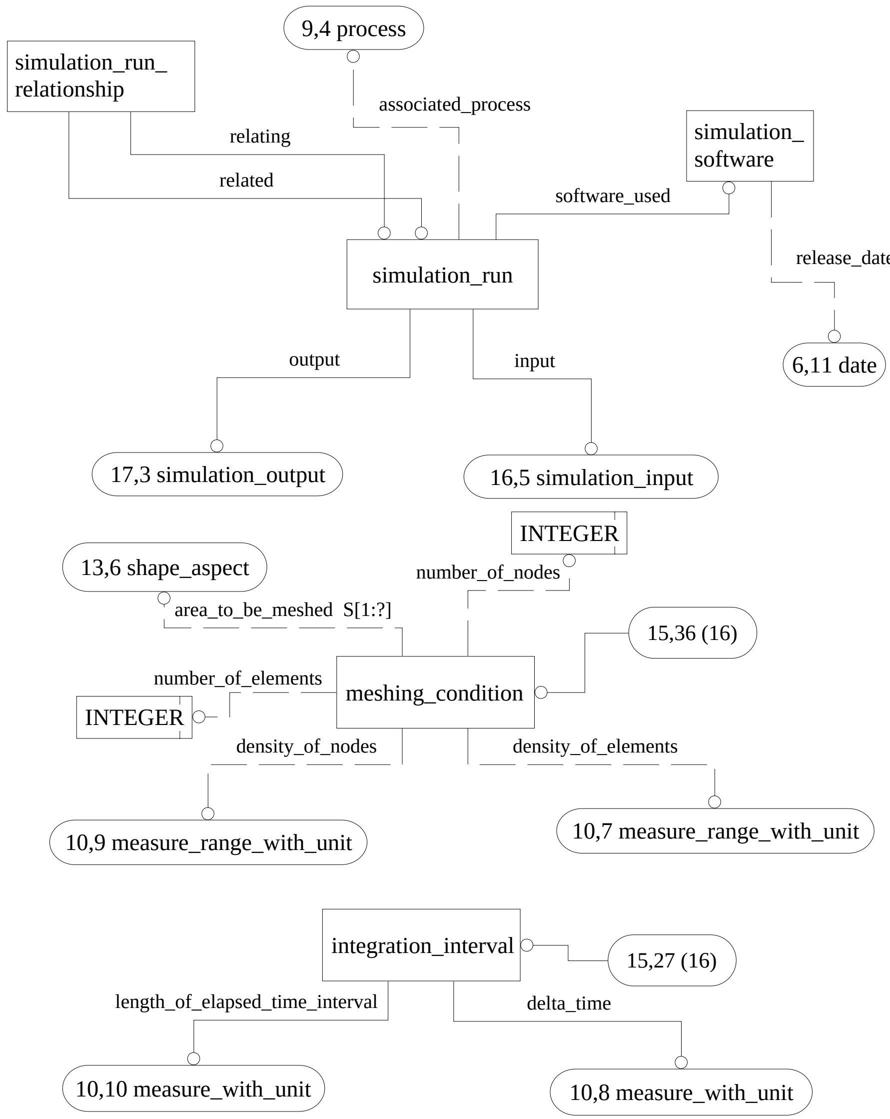


Figure G.15 – Simulation UoF

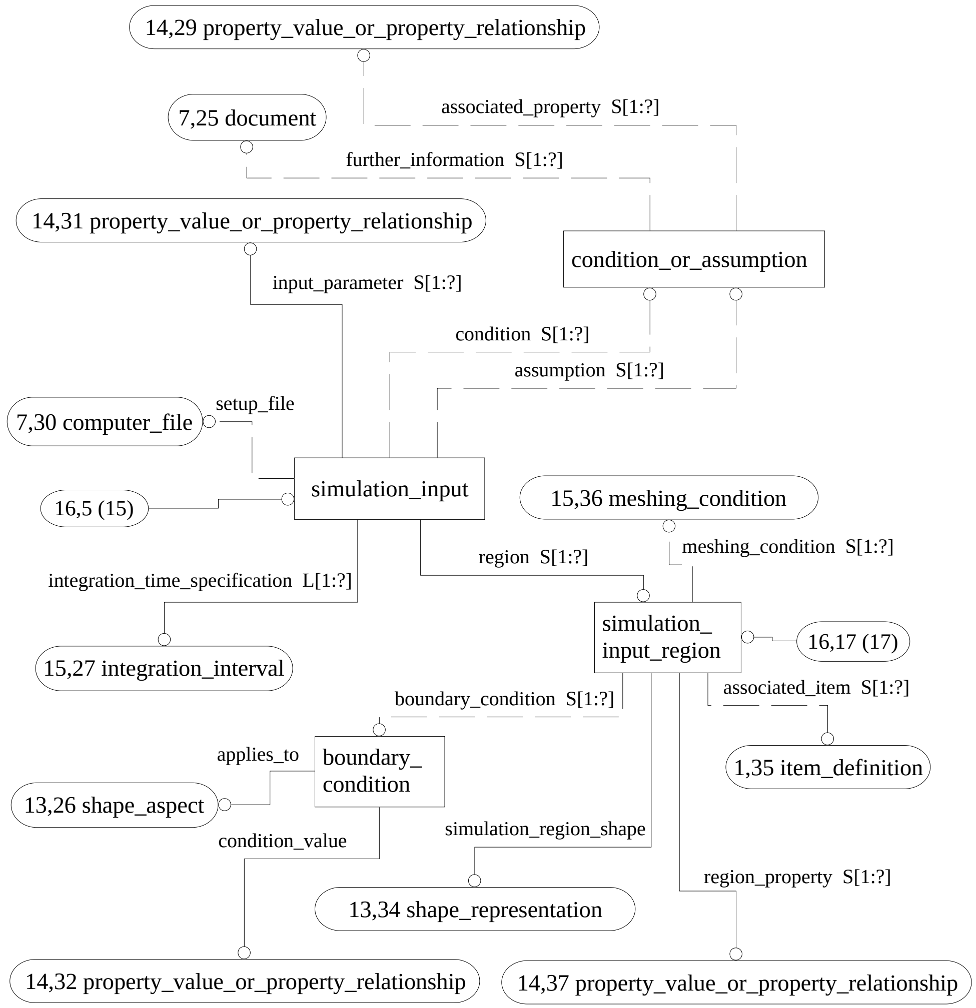


Figure G.16 – Simulation UoF

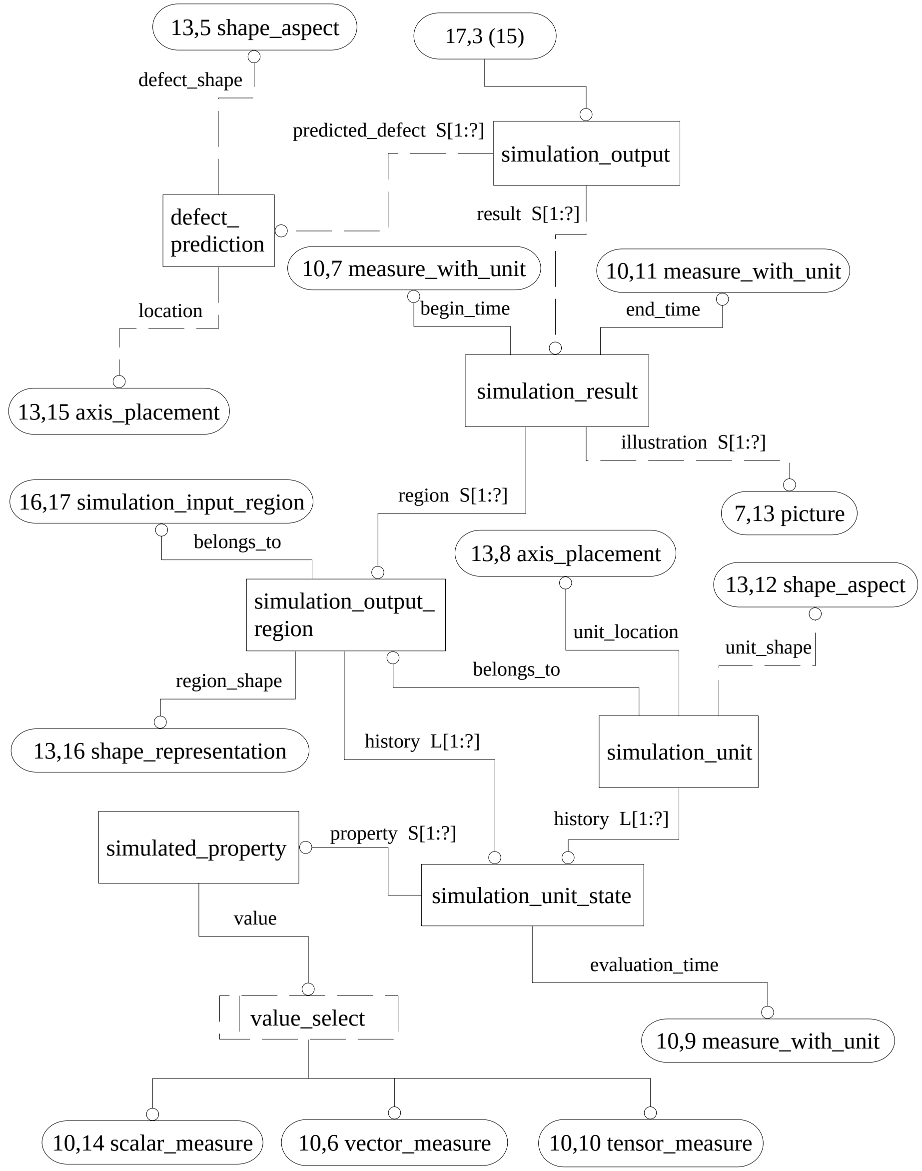


Figure G.17 – Simulation UoF

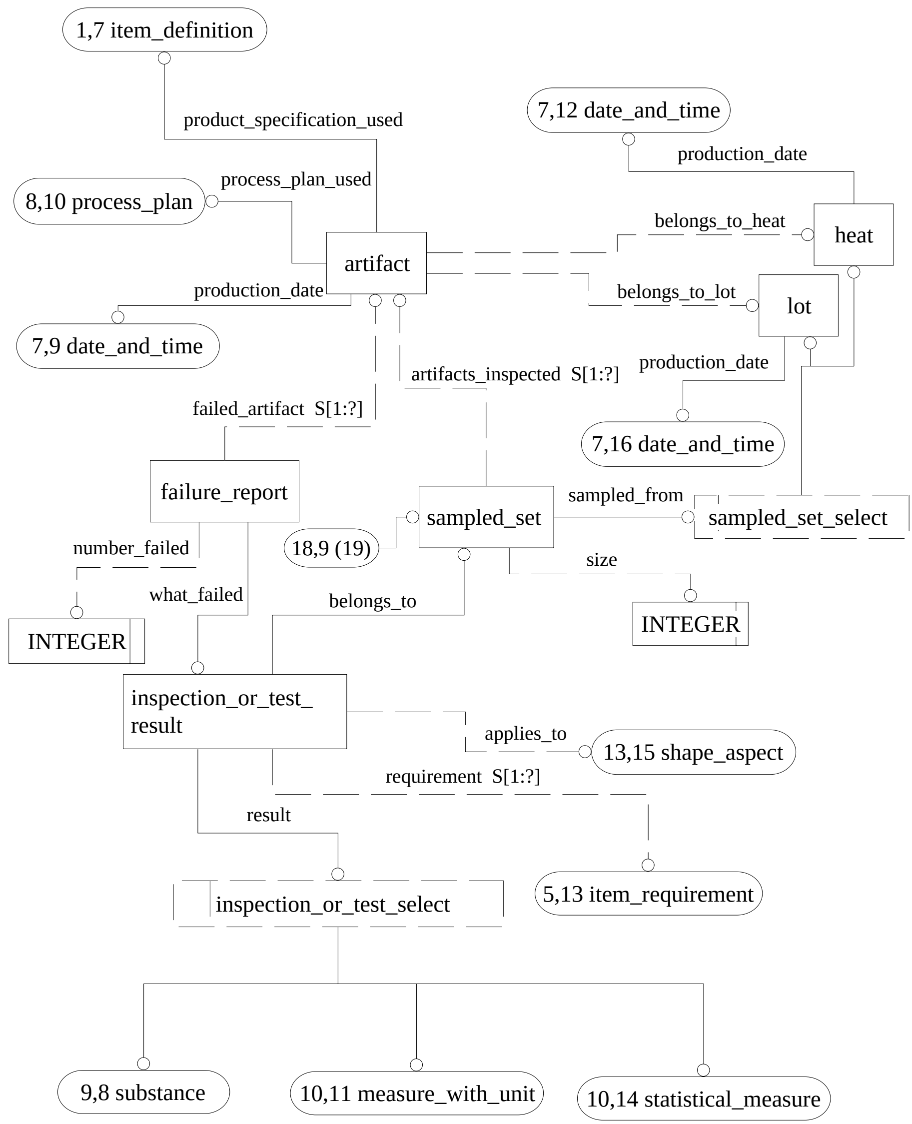


Figure G.18 – Quality Assurance UoF

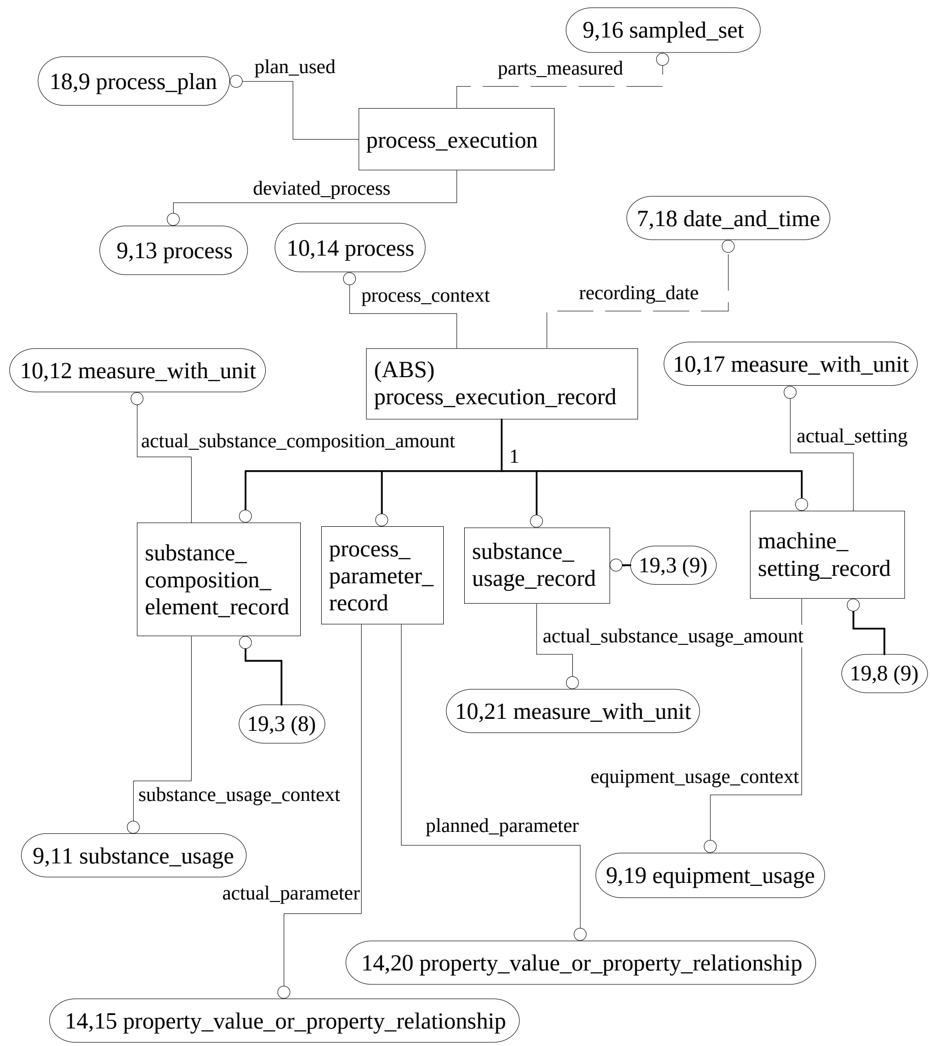


Figure G.19 – Quality Control UoF

Annex H
(informative)

AIM EXPRESS-G

Figures H.1 through H.34 correspond to the AIM EXPRESS expanded listing given in annex J. The figures use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex A of ISO 10303-11.

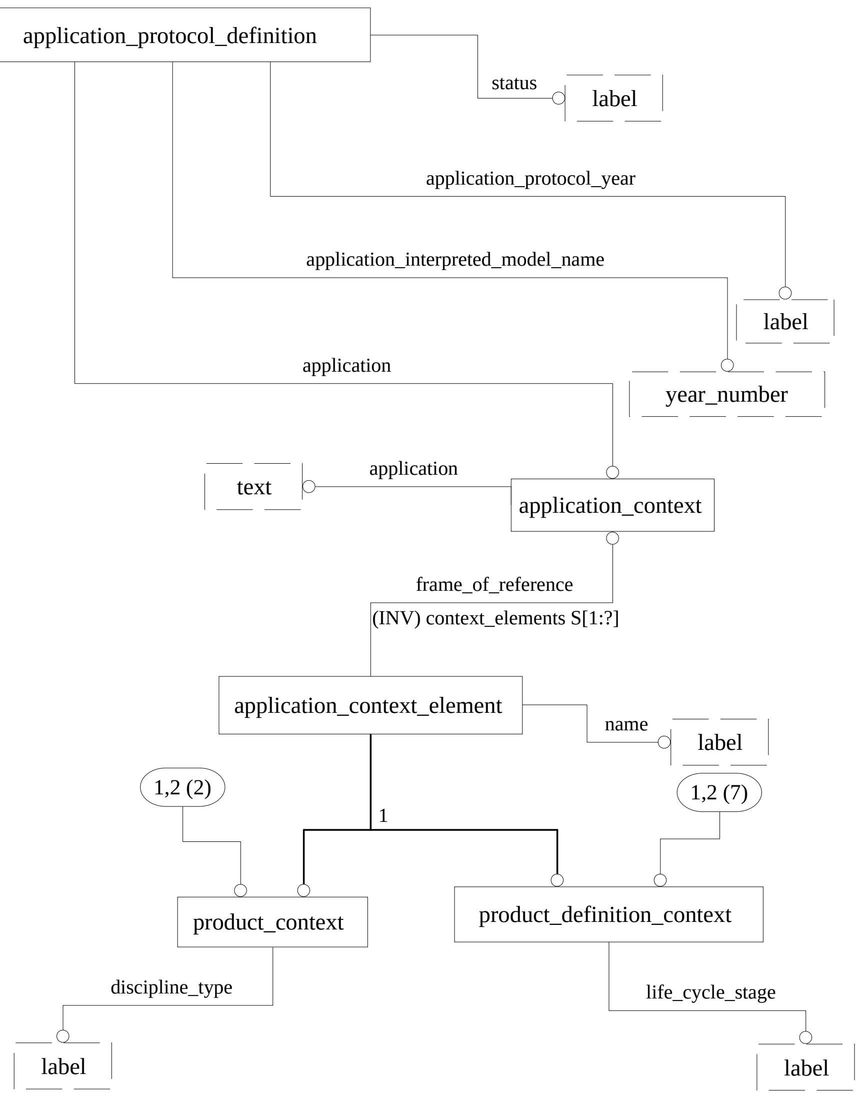


Figure H.1 – application context - AIM EXPRESS-G diagram 1 of 34

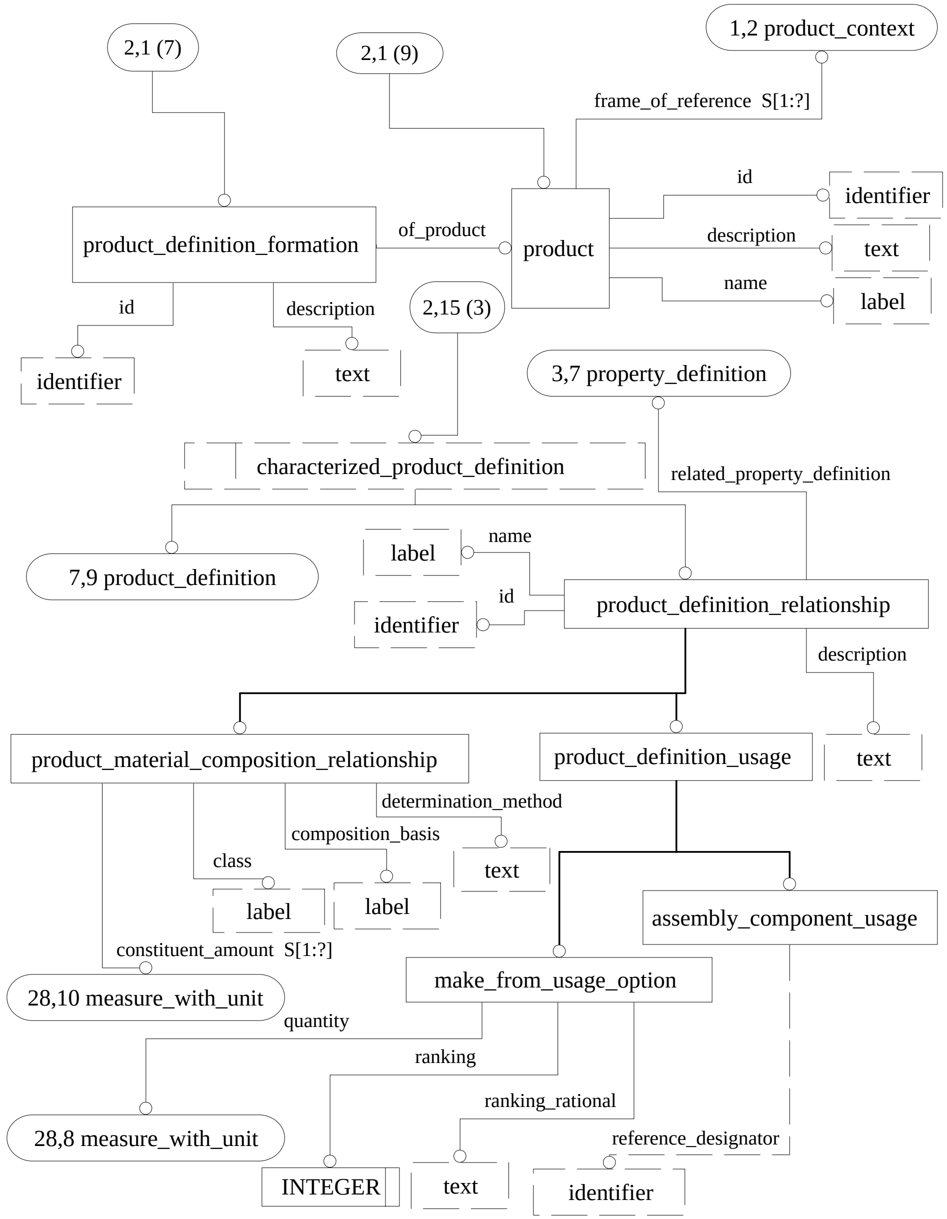


Figure H.2 – product definition - AIM EXPRESS-G diagram 2 of 34

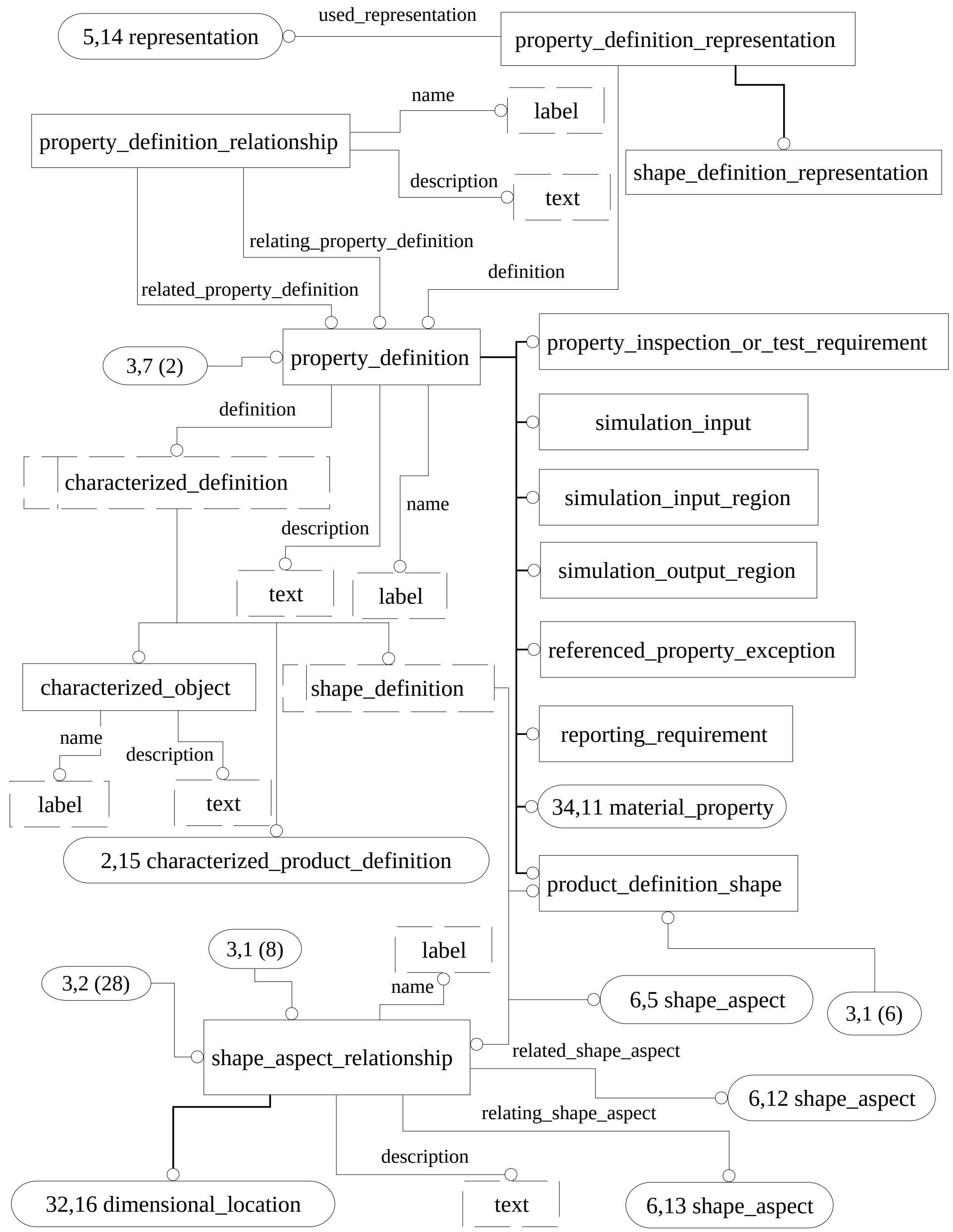


Figure H.3 – property definition - AIM EXPRESS-G diagram 3 of 34

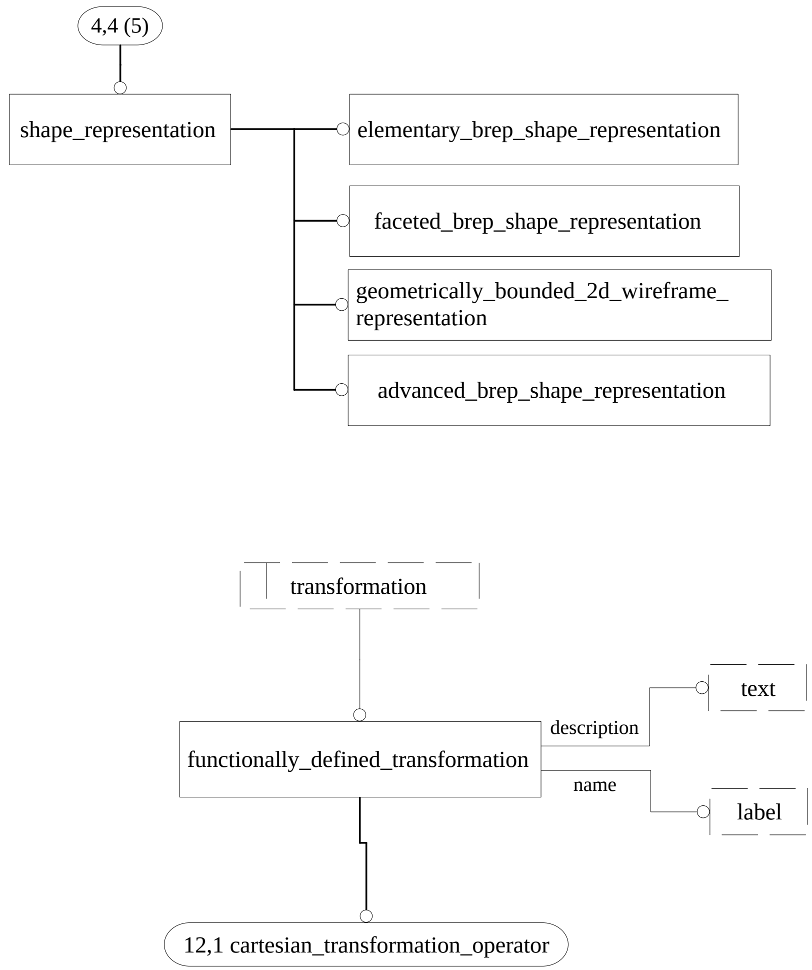


Figure H.4 – shape representation - AIM EXPRESS-G diagram 4 of 34

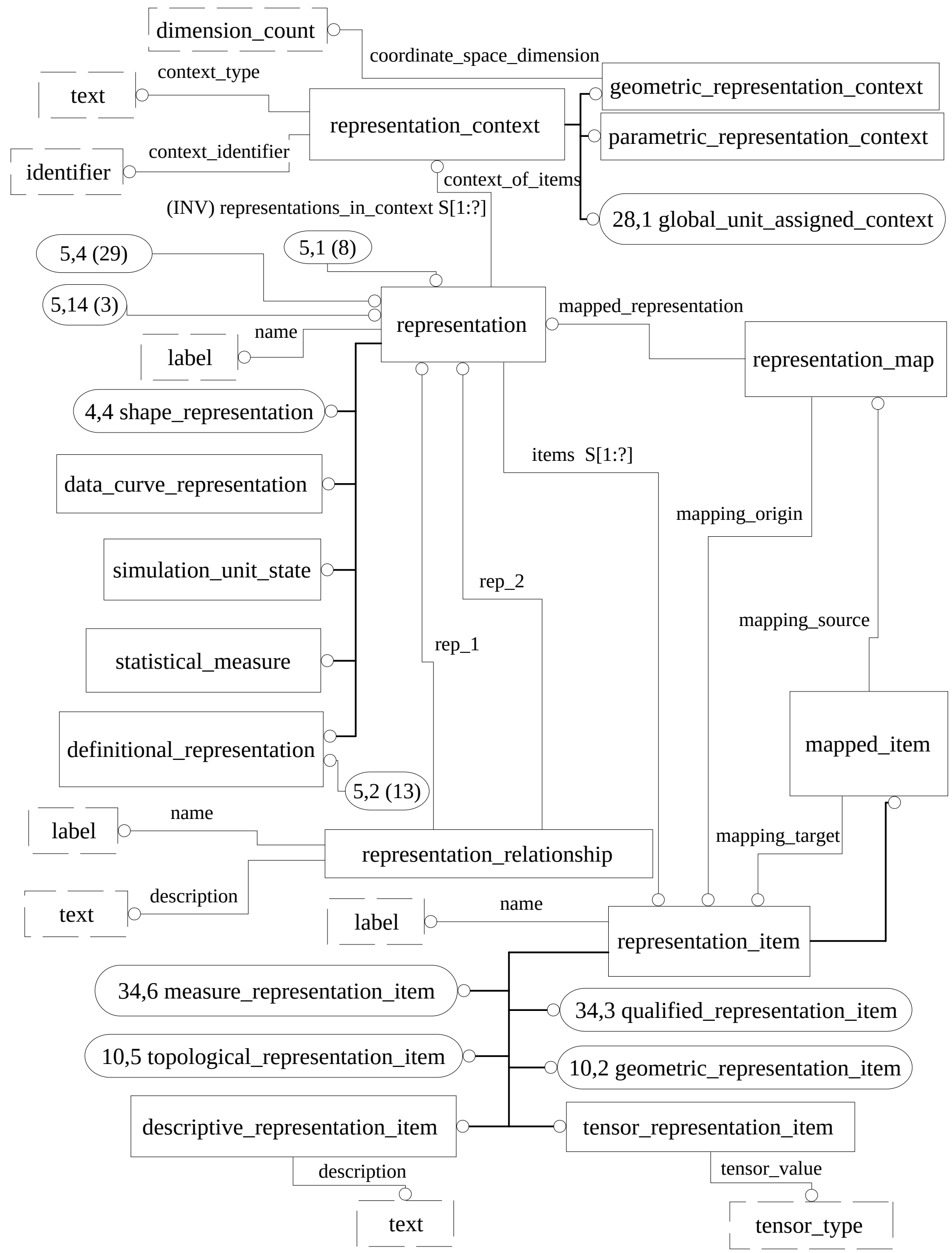


Figure H.5 – representation - AIM EXPRESS-G diagram 5 of 34

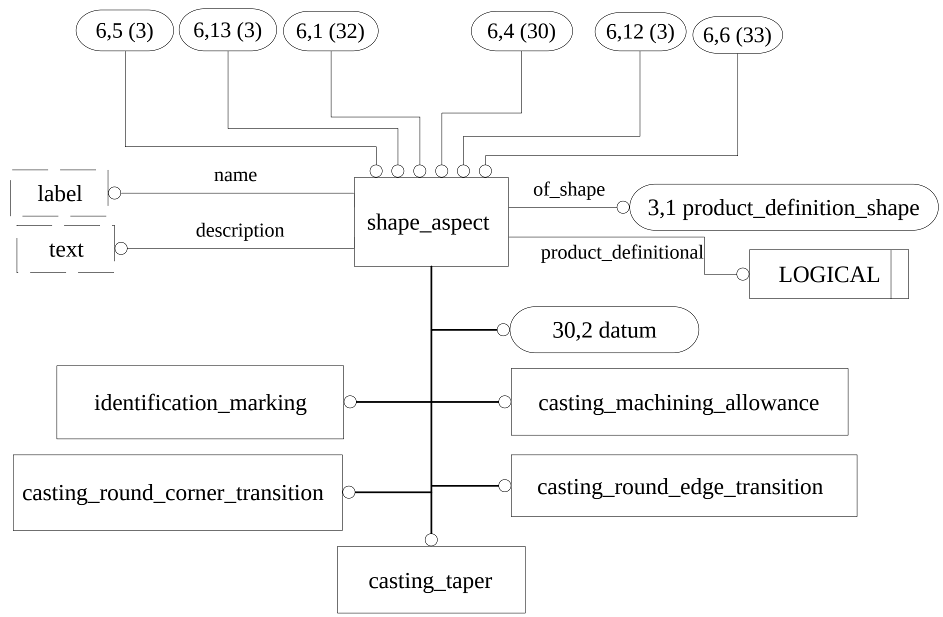


Figure H.6 – shape aspect - AIM EXRESS-G - AIM EXPRESS-G diagram 6 of 34

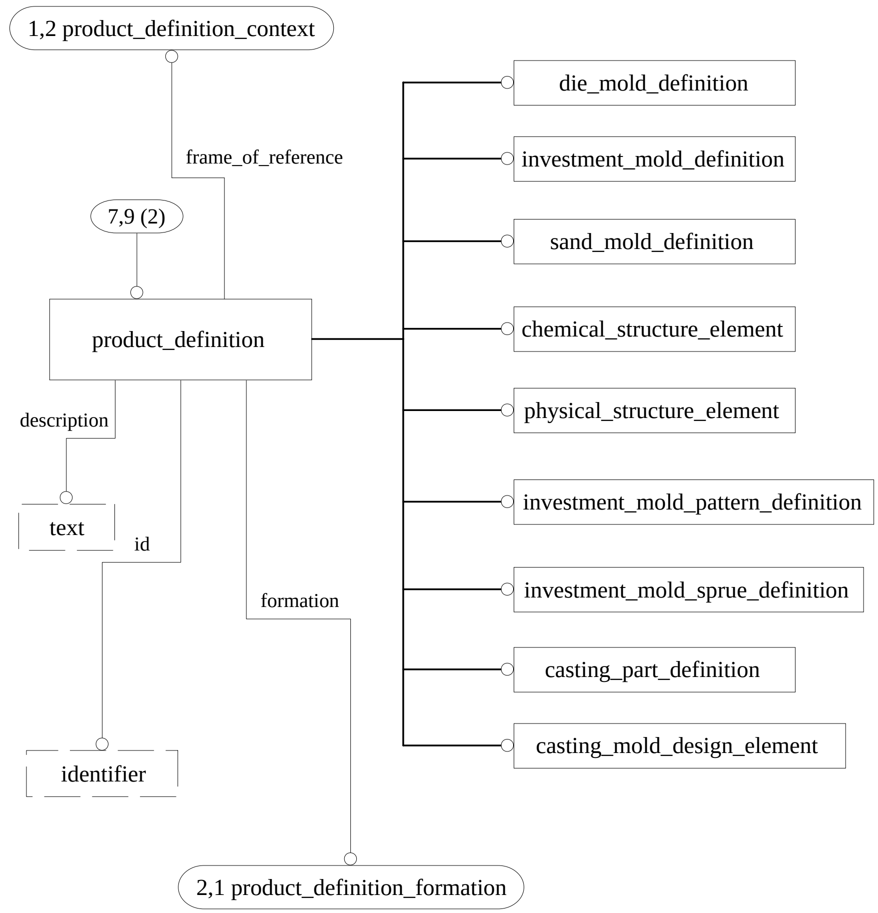


Figure H.7 – casting feature - AIM EXPRES - AIM EXPRESS-G diagram 7 of 34

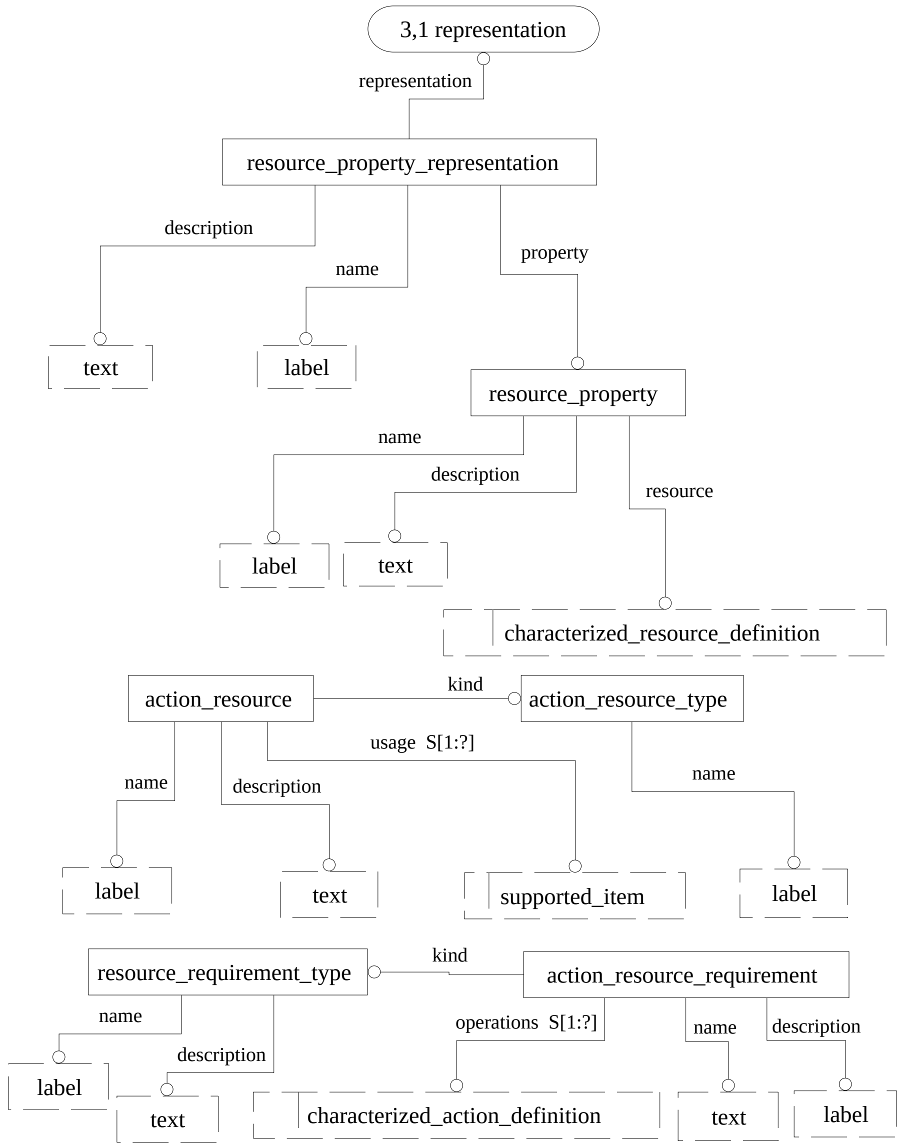


Figure H.8 – resource description - AIM EXPRESS-G diagram 8 of 34

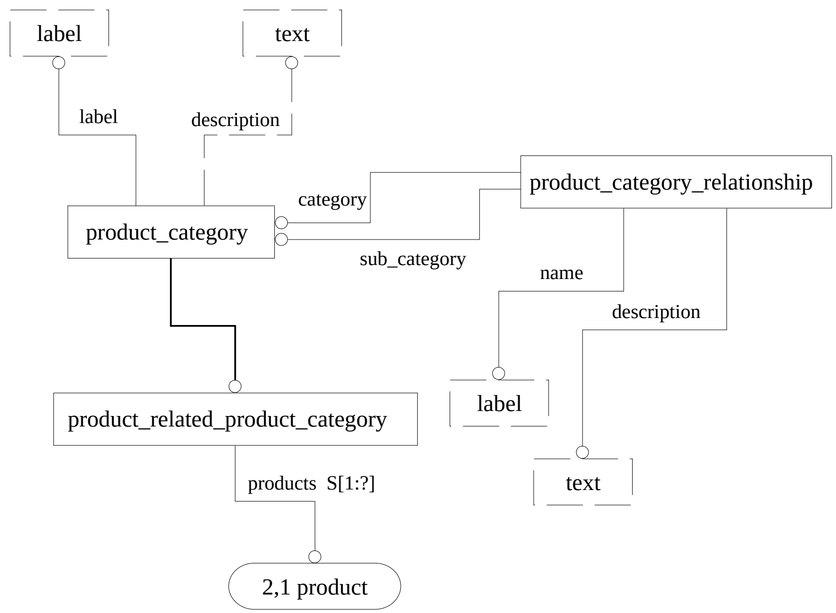


Figure H.9 – product category - AIM EXPRESS-G diagram 9 of 34

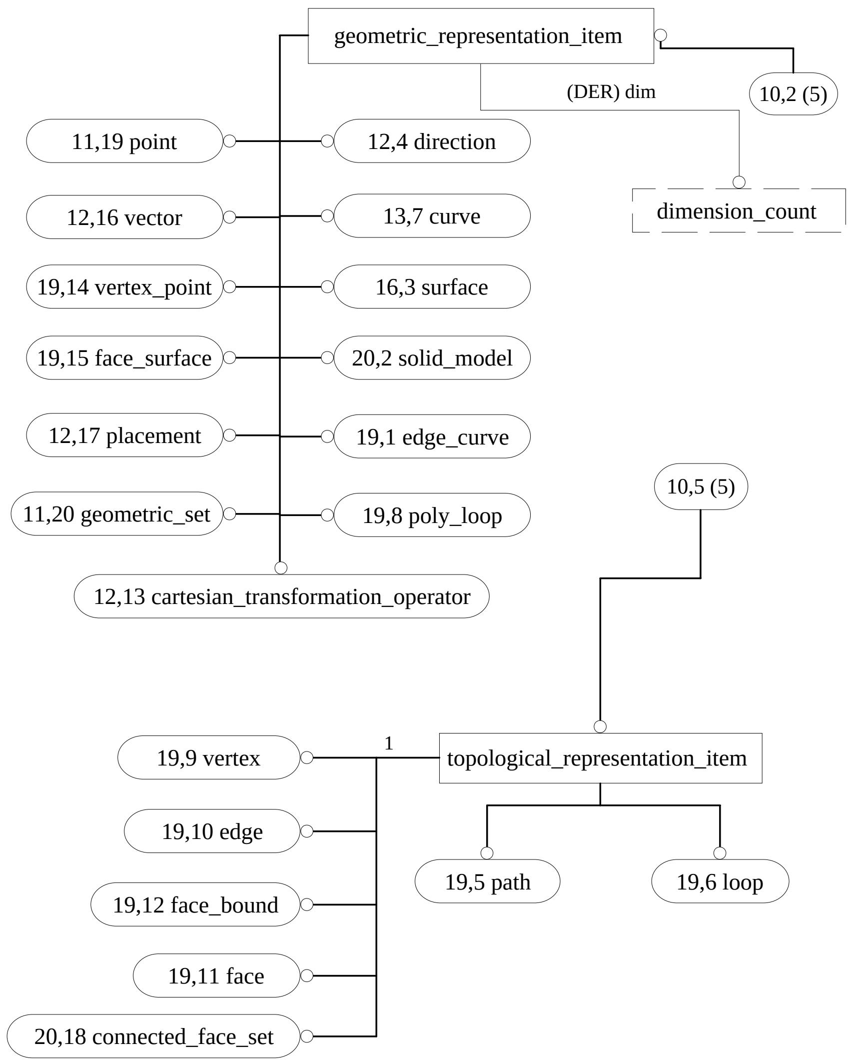


Figure H.10 – geometry topology - AIM EXPRESS-G diagram 10 of 34

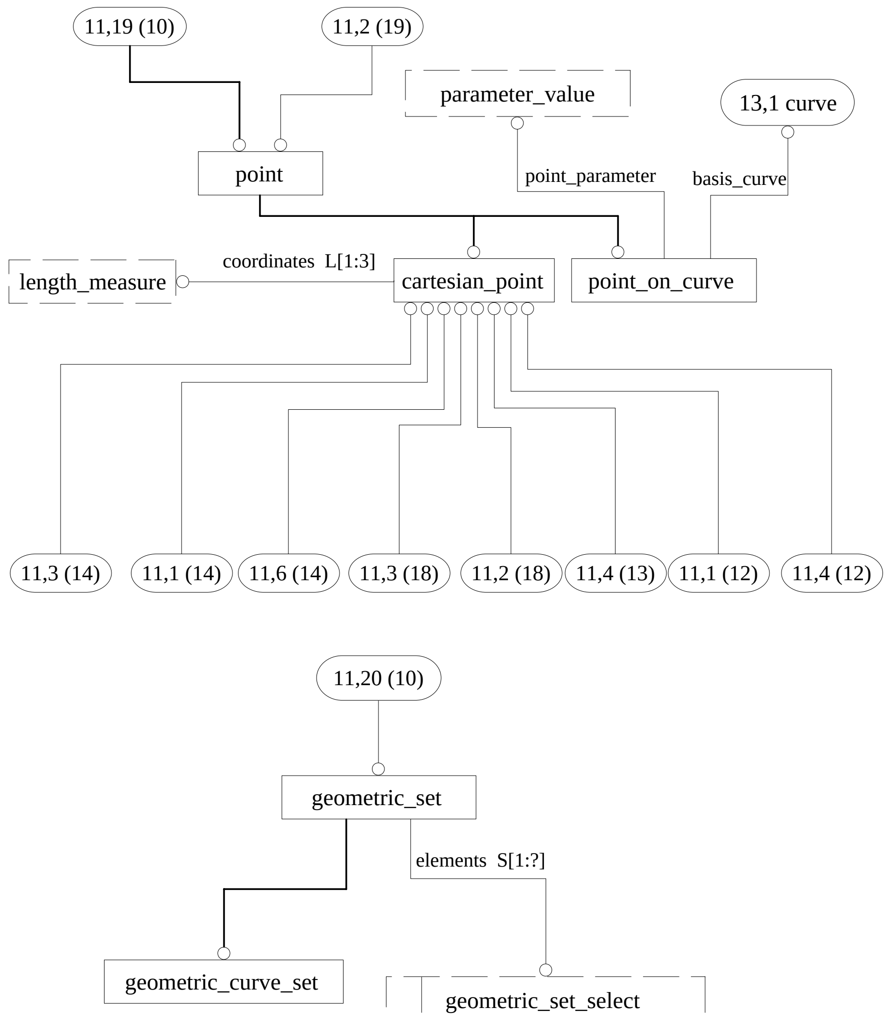


Figure H.11 – point - AIM EXPRESS-G diagram 11 of 34

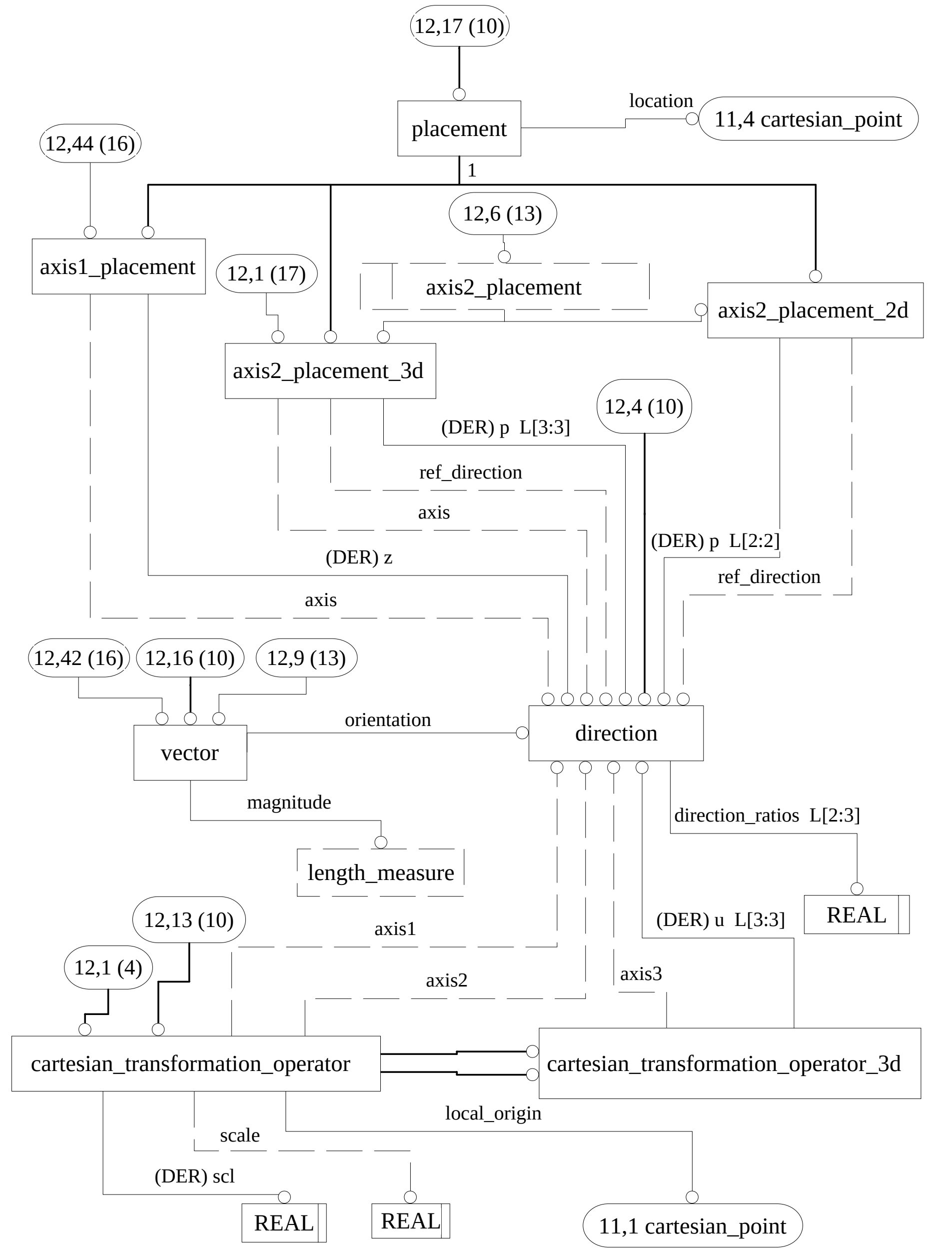


Figure H.12 – geometric orientation - AIM EXPRESS-G diagram 12 of 34

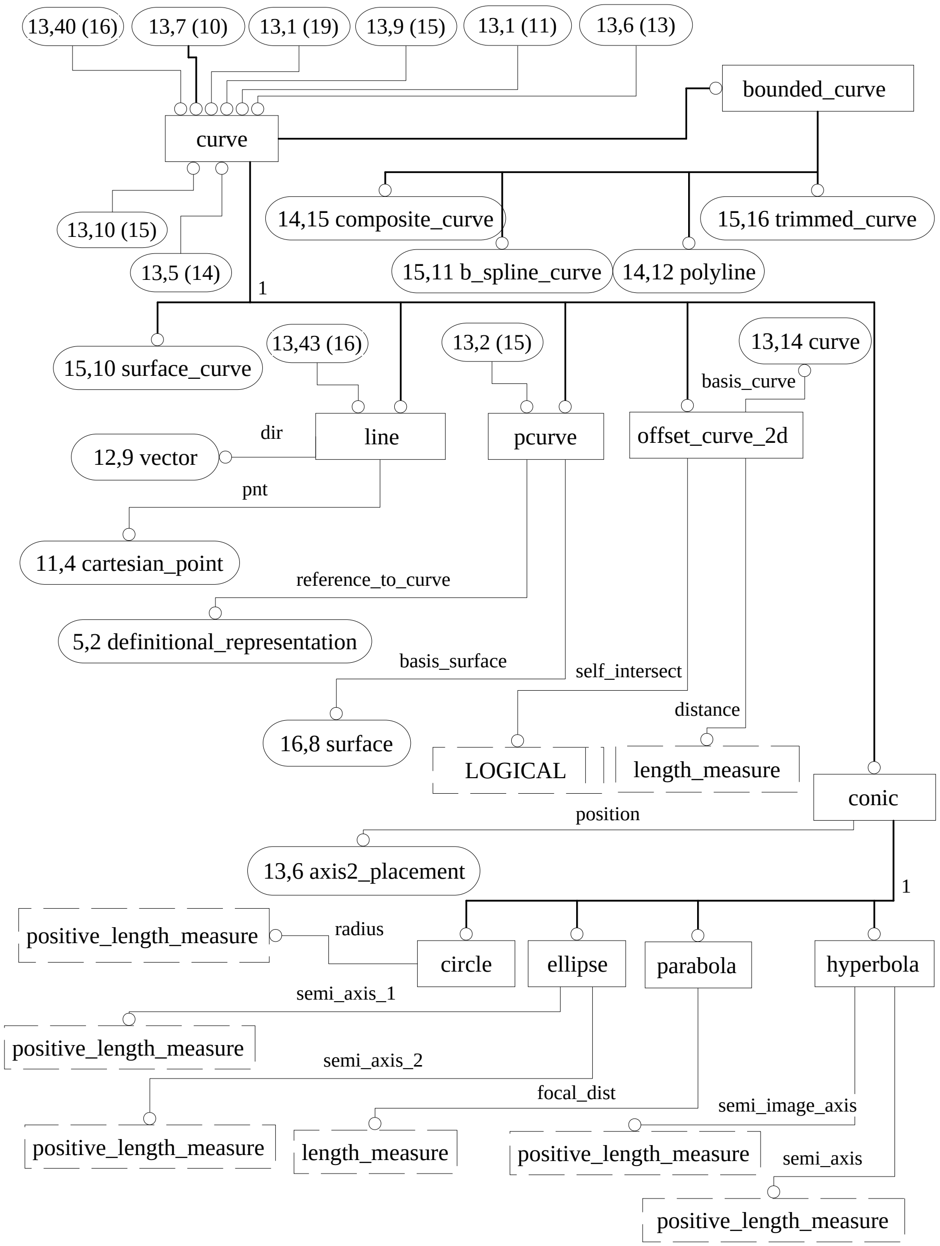


Figure H.13 – curve - AIM EXPRESS-G diagram 13 of 34

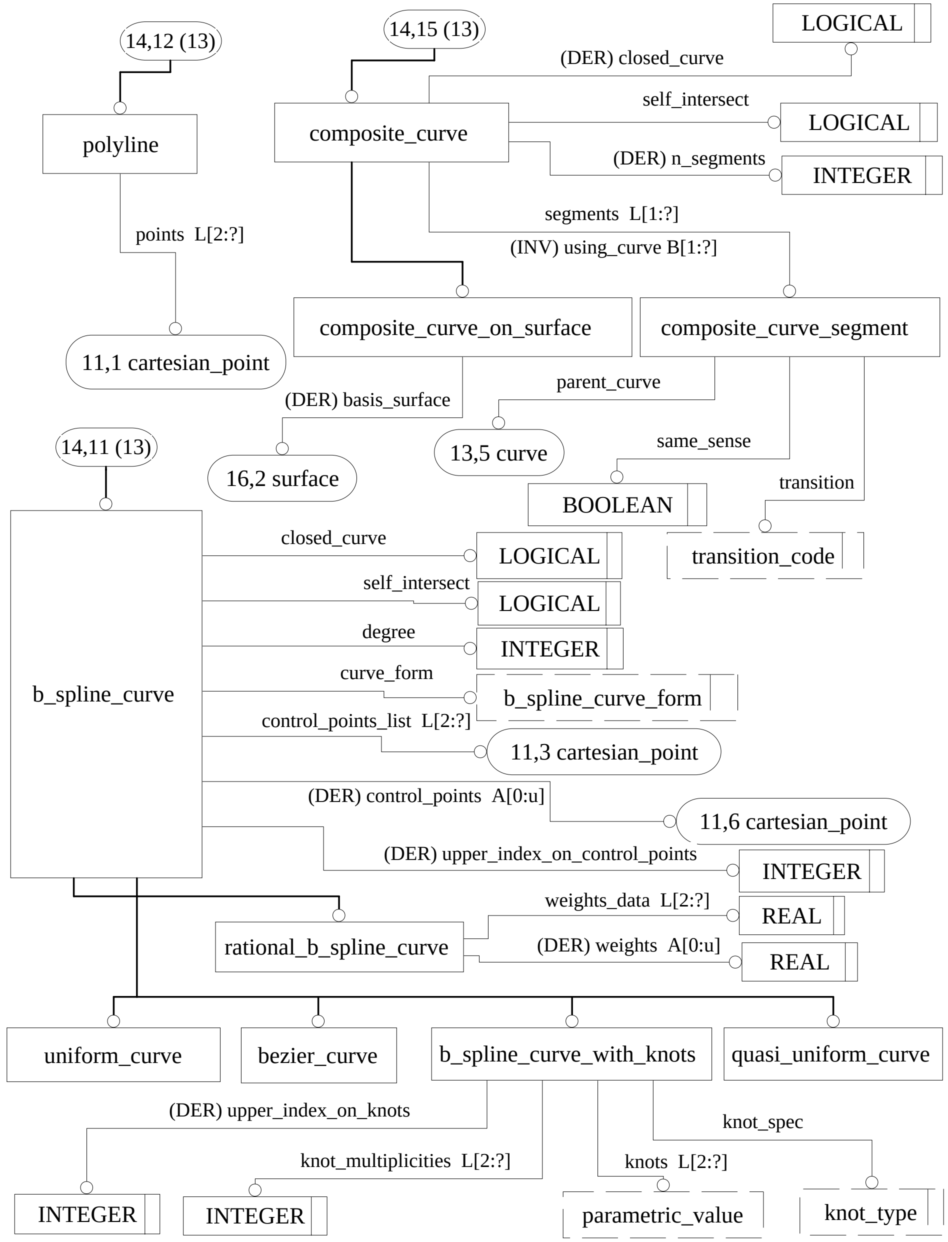


Figure H.14 – b_spline curve - AIM EXPRESS-G diagram 14 of 34

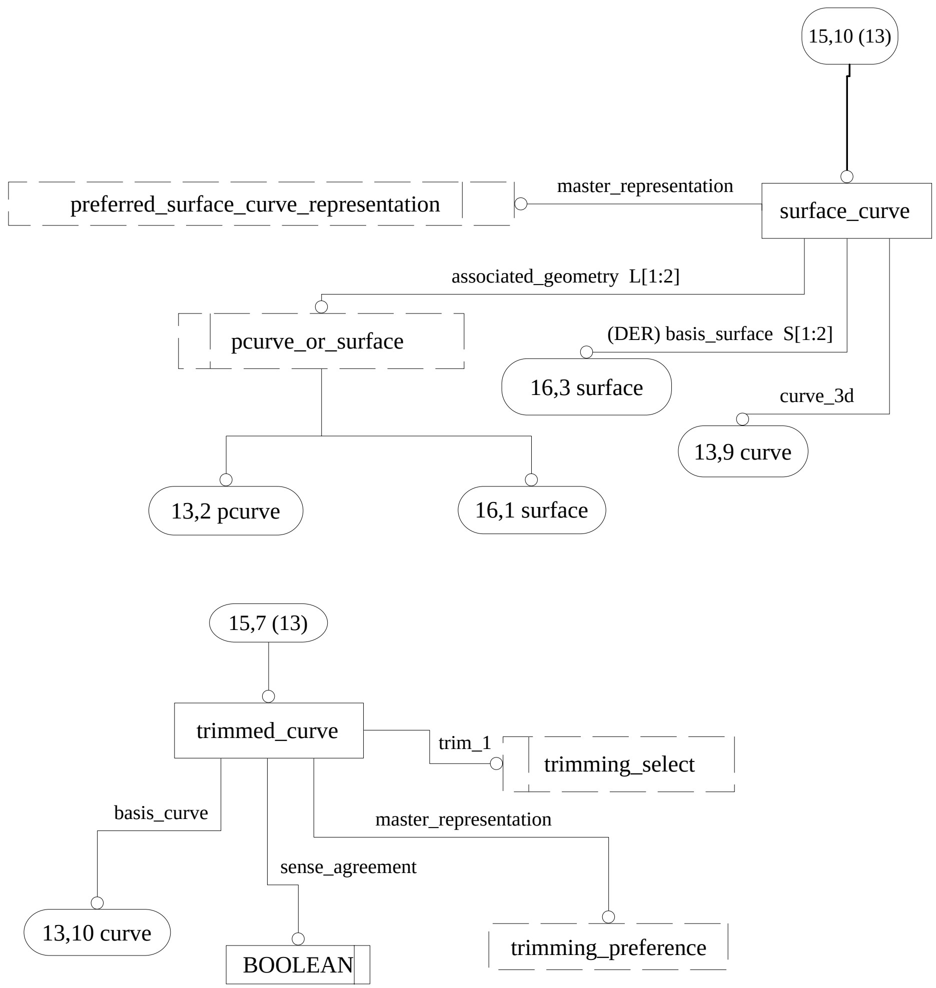


Figure H.15 – surface curve - AIM EXPRESS-G diagram 15 of 34

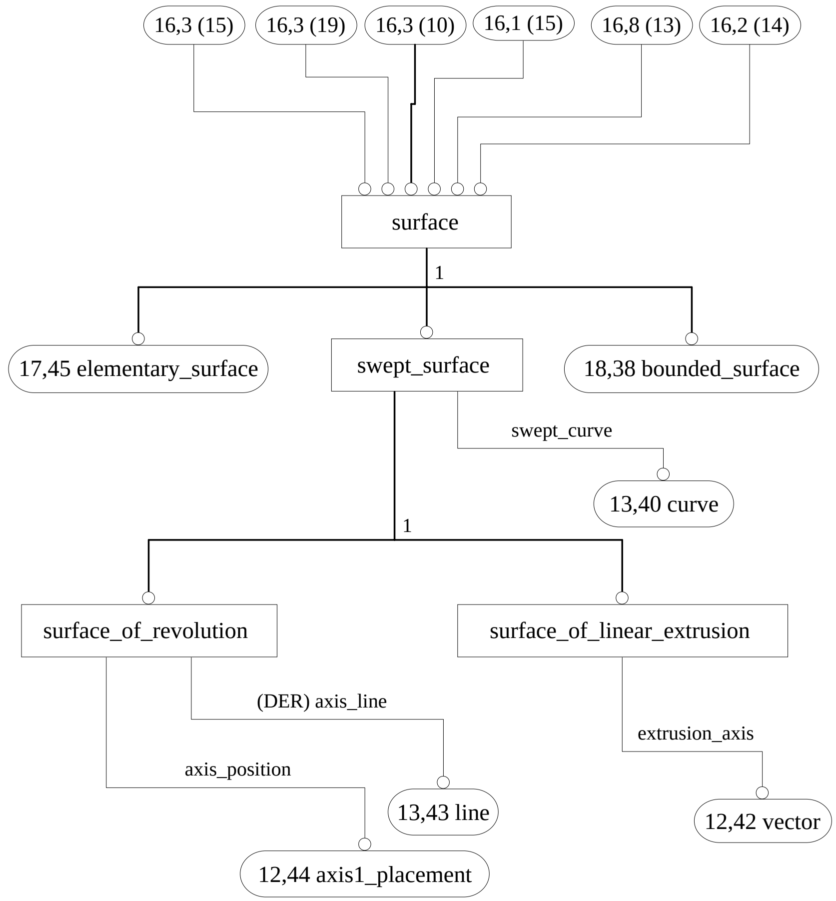


Figure H.16 – surface - AIM EXPRESS-G diagram 16 of 34

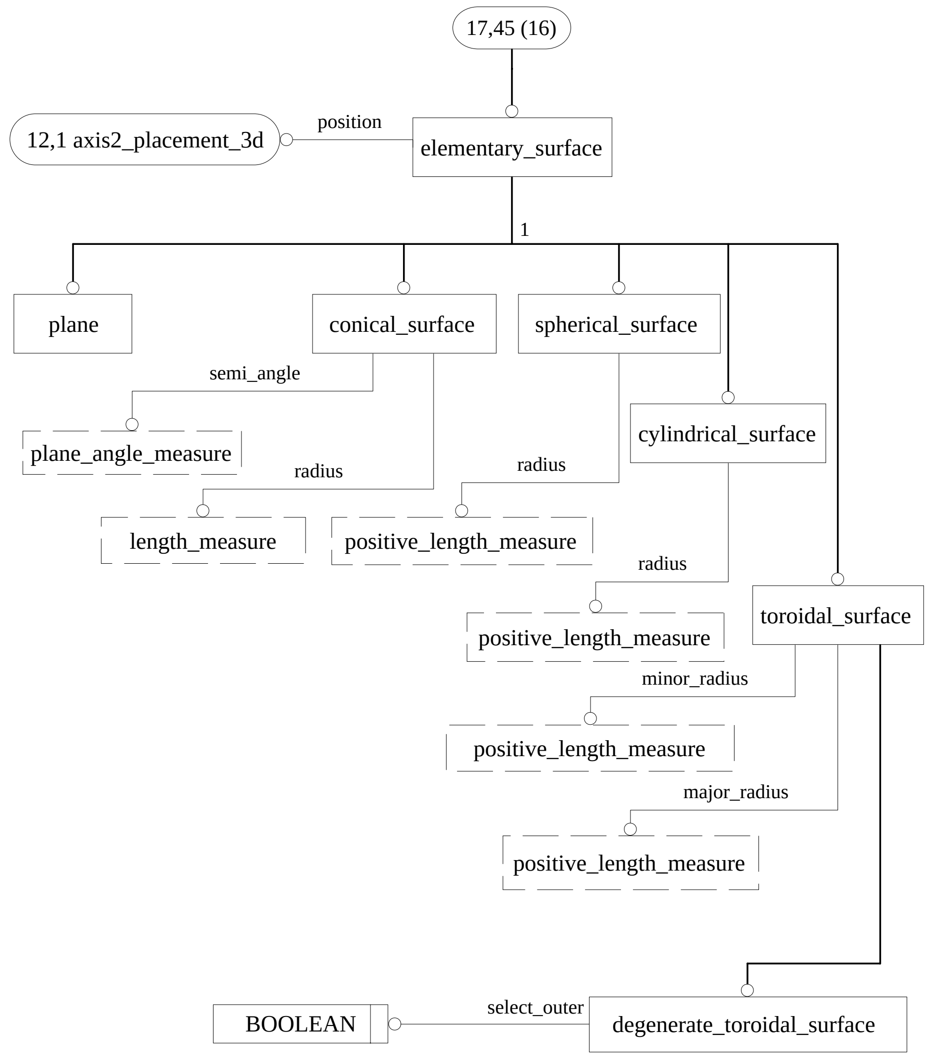


Figure H.17 – elementary surface - AIM EXPRESS-G diagram 17 of 34

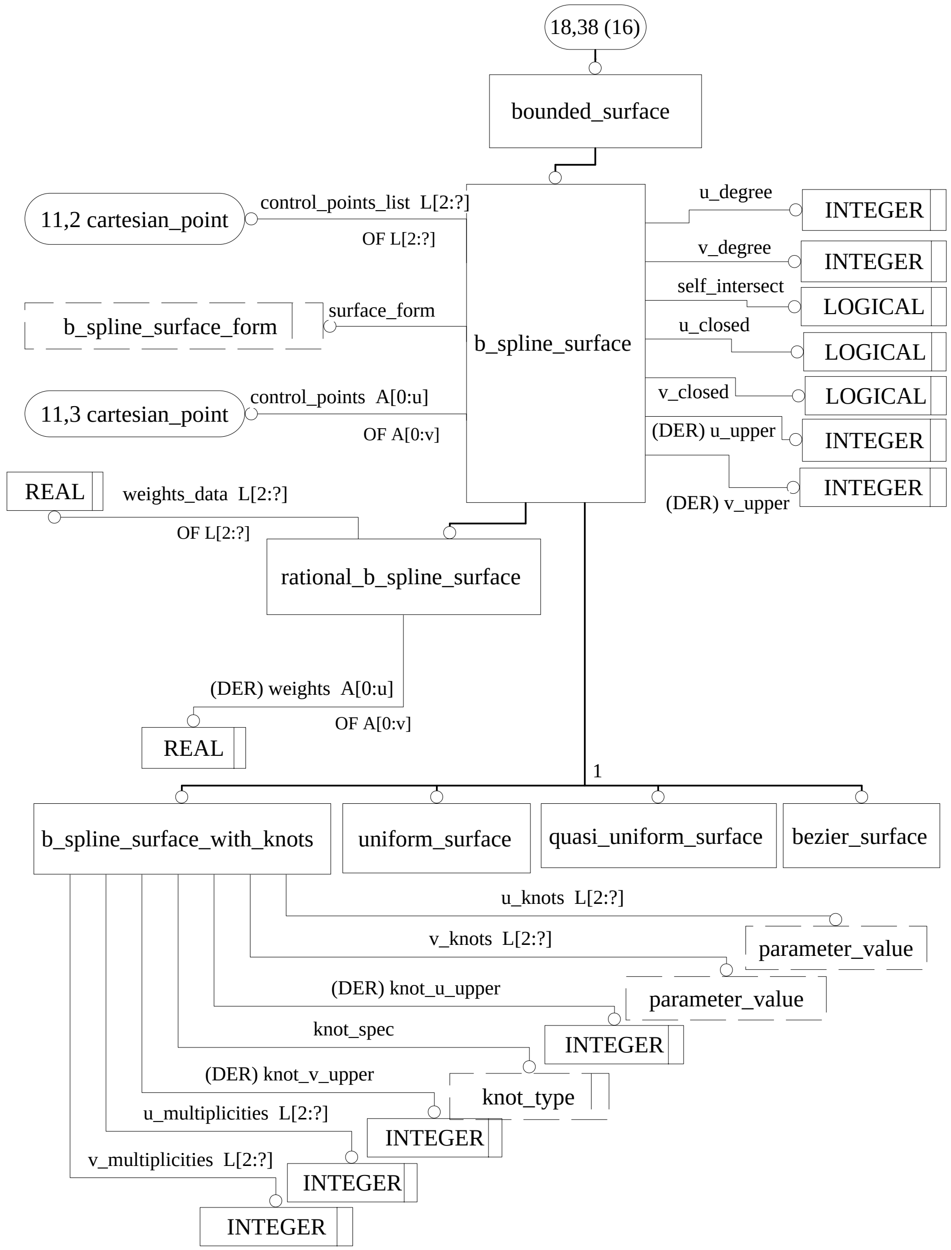


Figure H.18 – b_spline surface - AIM EXPRESS-G diagram 18 of 34

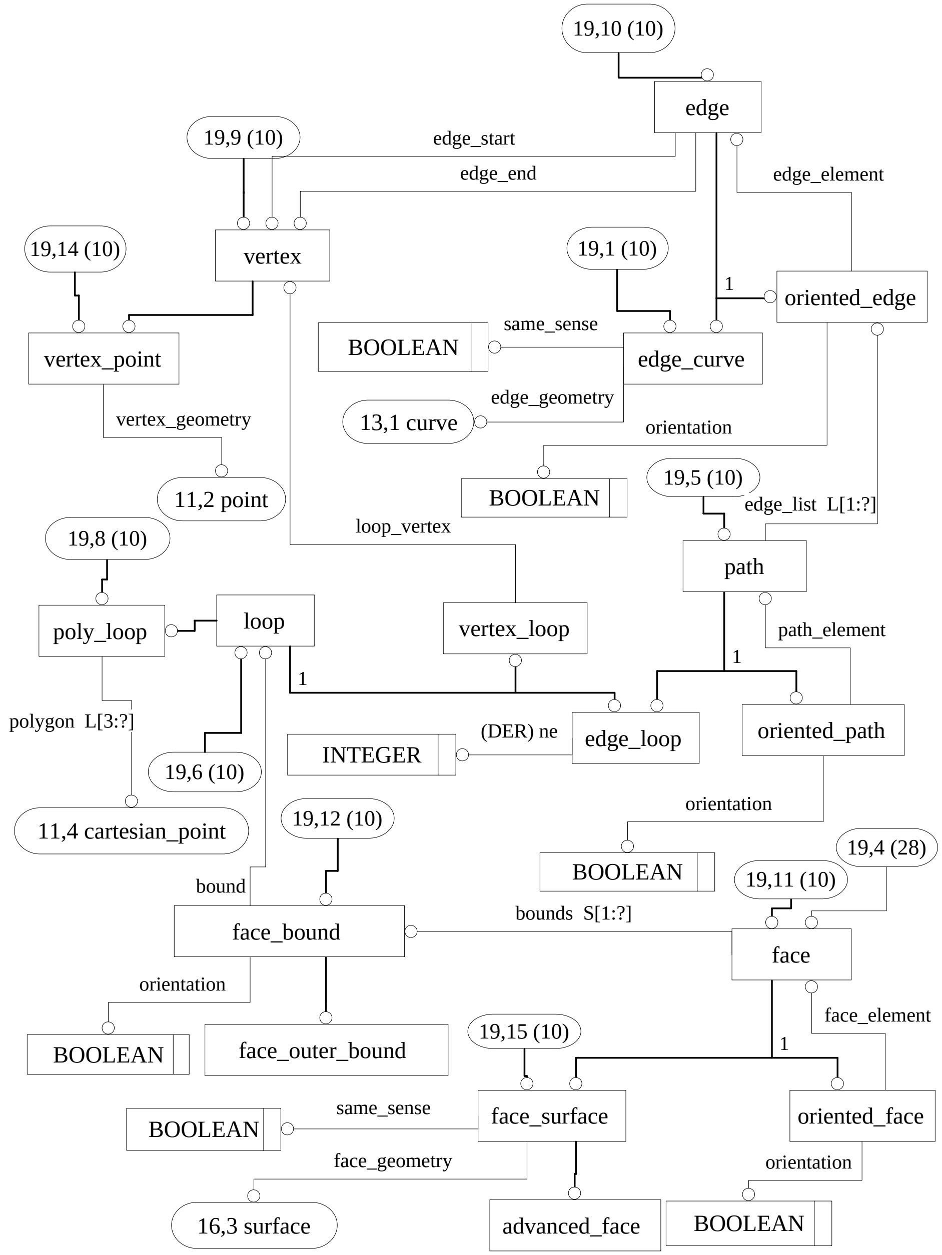


Figure H.19 – topology - AIM EXPRESS-G diagram 19 of 34

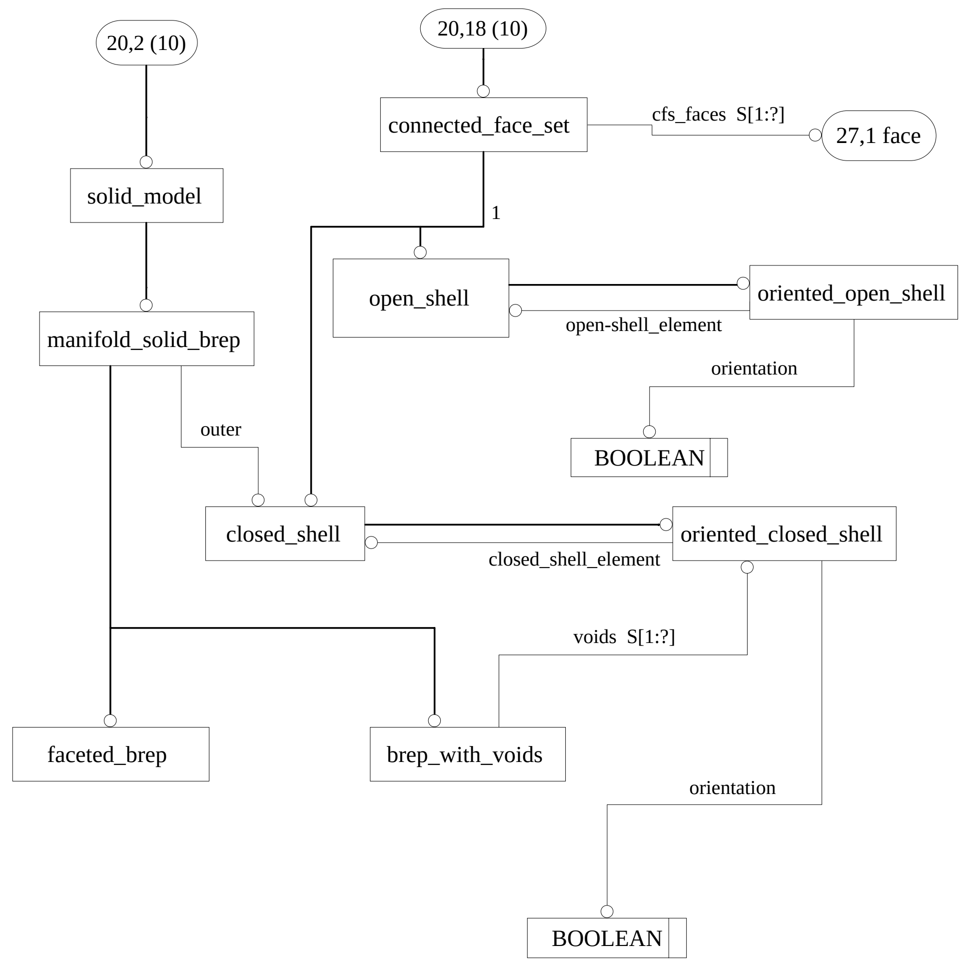


Figure H.20 – shell - AIM EXPRESS-G diagram 20 of 34

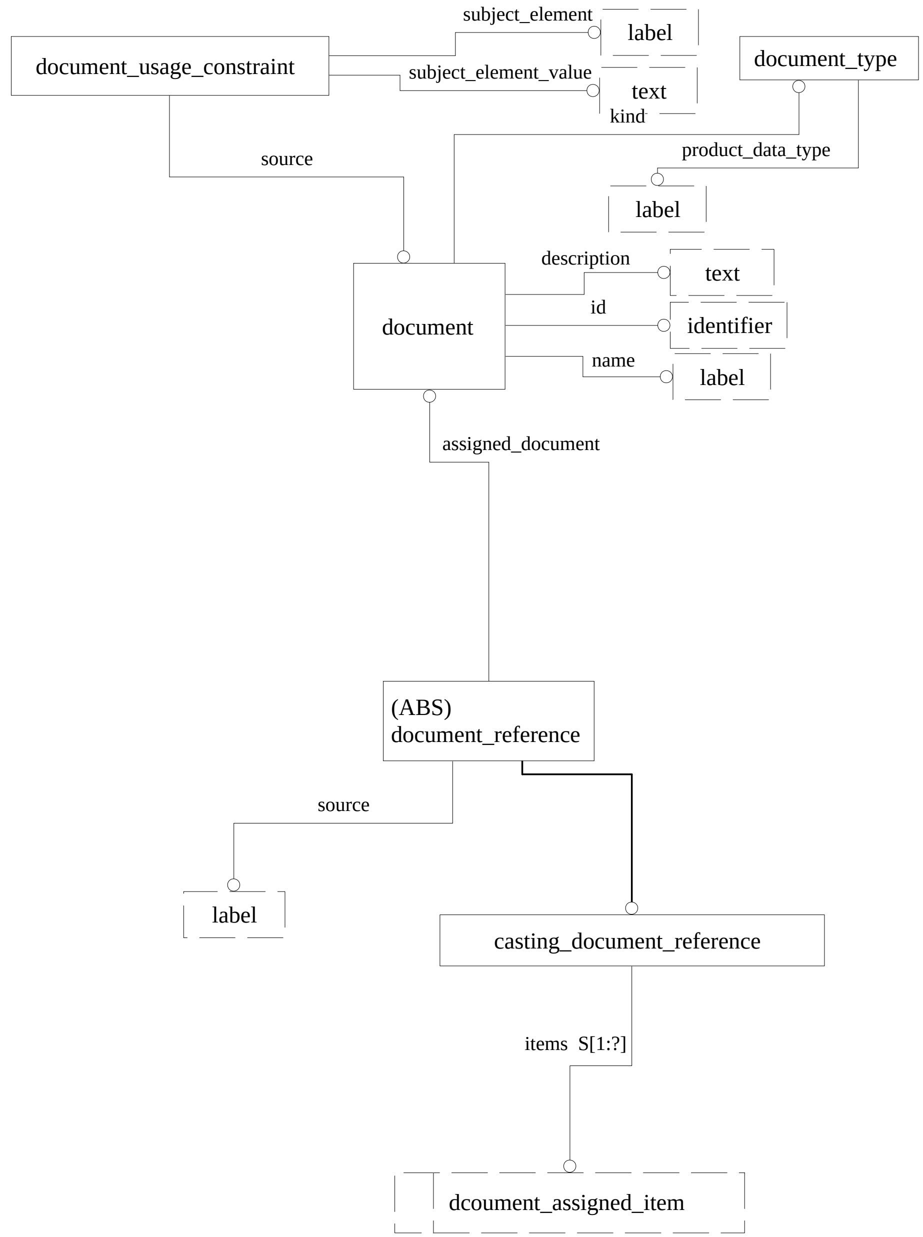


Figure H.21 – document - AIM EXPRESS-G diagram 21 of 34

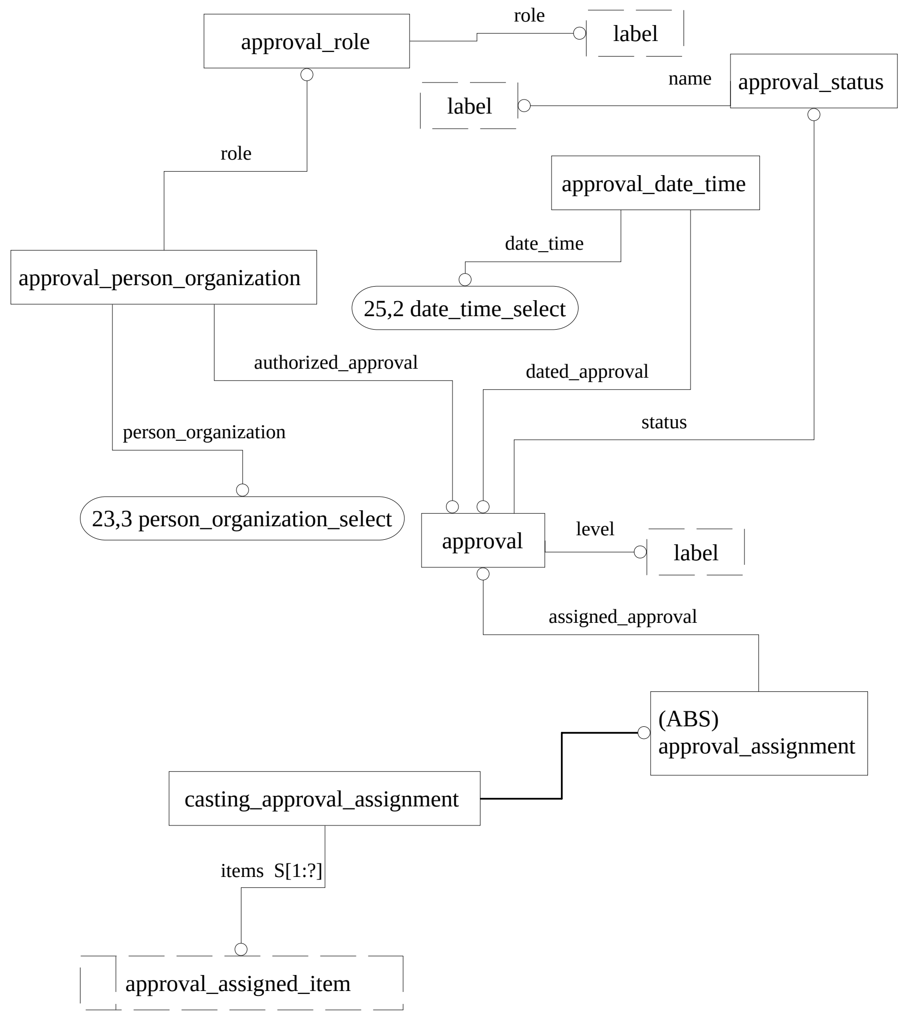


Figure H.22 – approval - AIM EXPRESS-G diagram 22 of 34

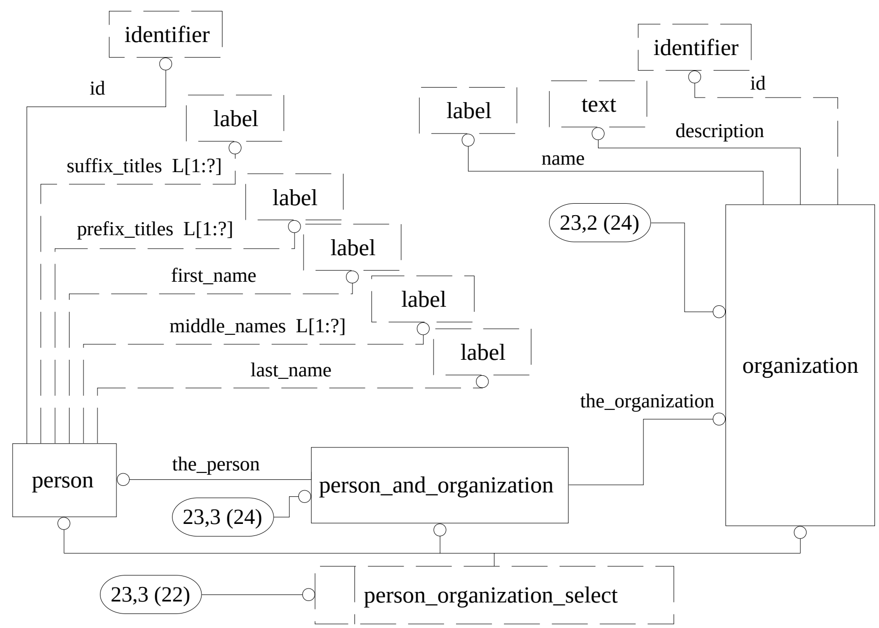


Figure H.23 – person and organization - AIM EXPRESS-G diagram 23 of 34

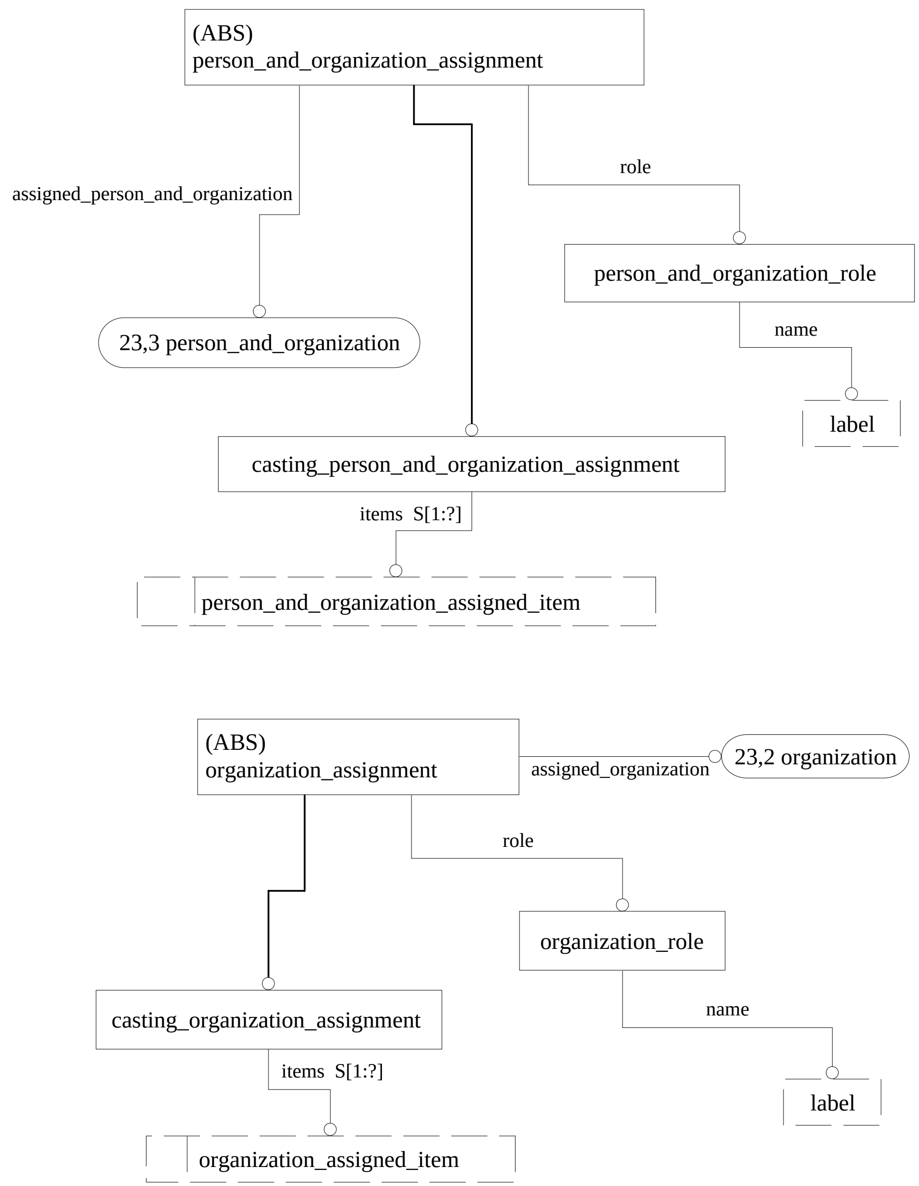


Figure H.24 – person and organization assignment - AIM EXPRESS-G diagram 24 of 34

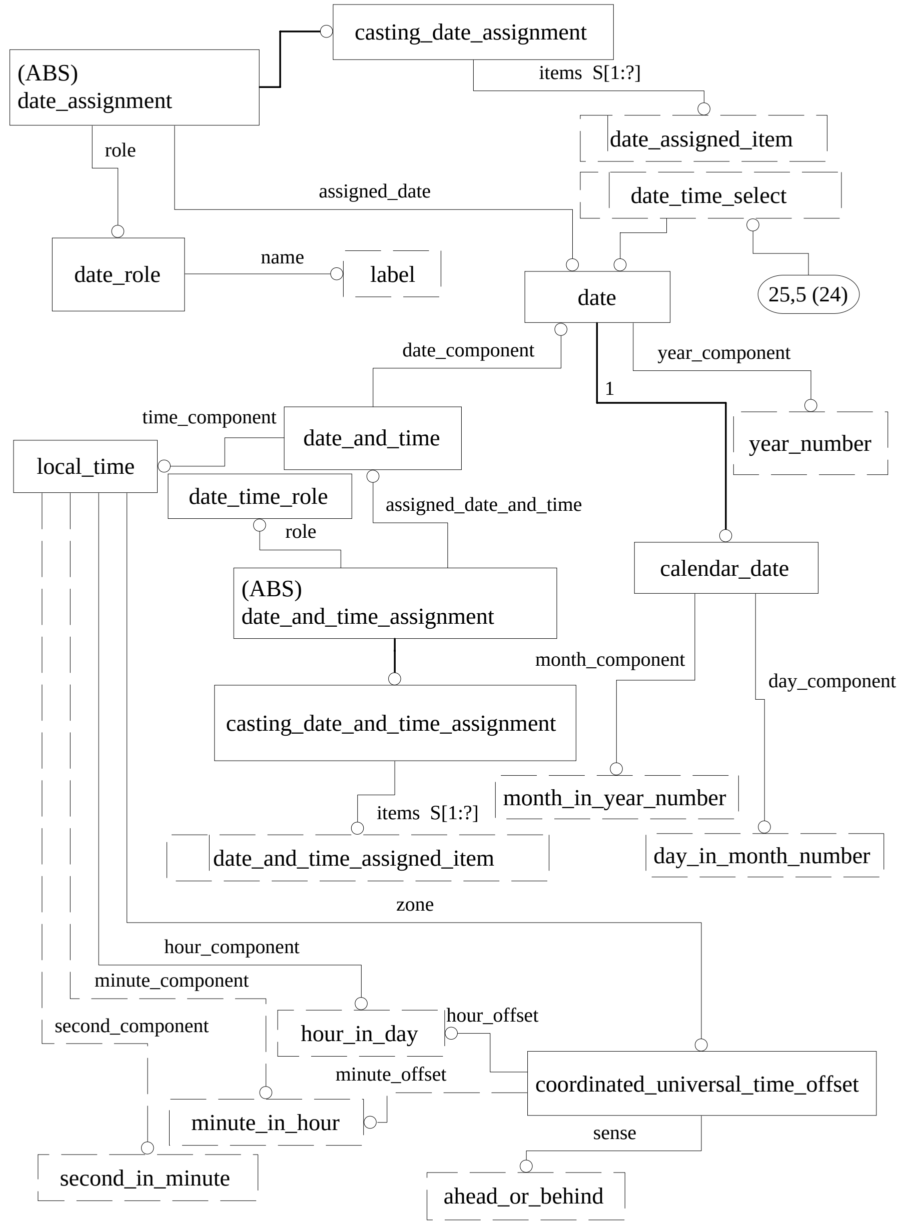


Figure H.25 – date - AIM EXPRESS-G diagram 25 of 34

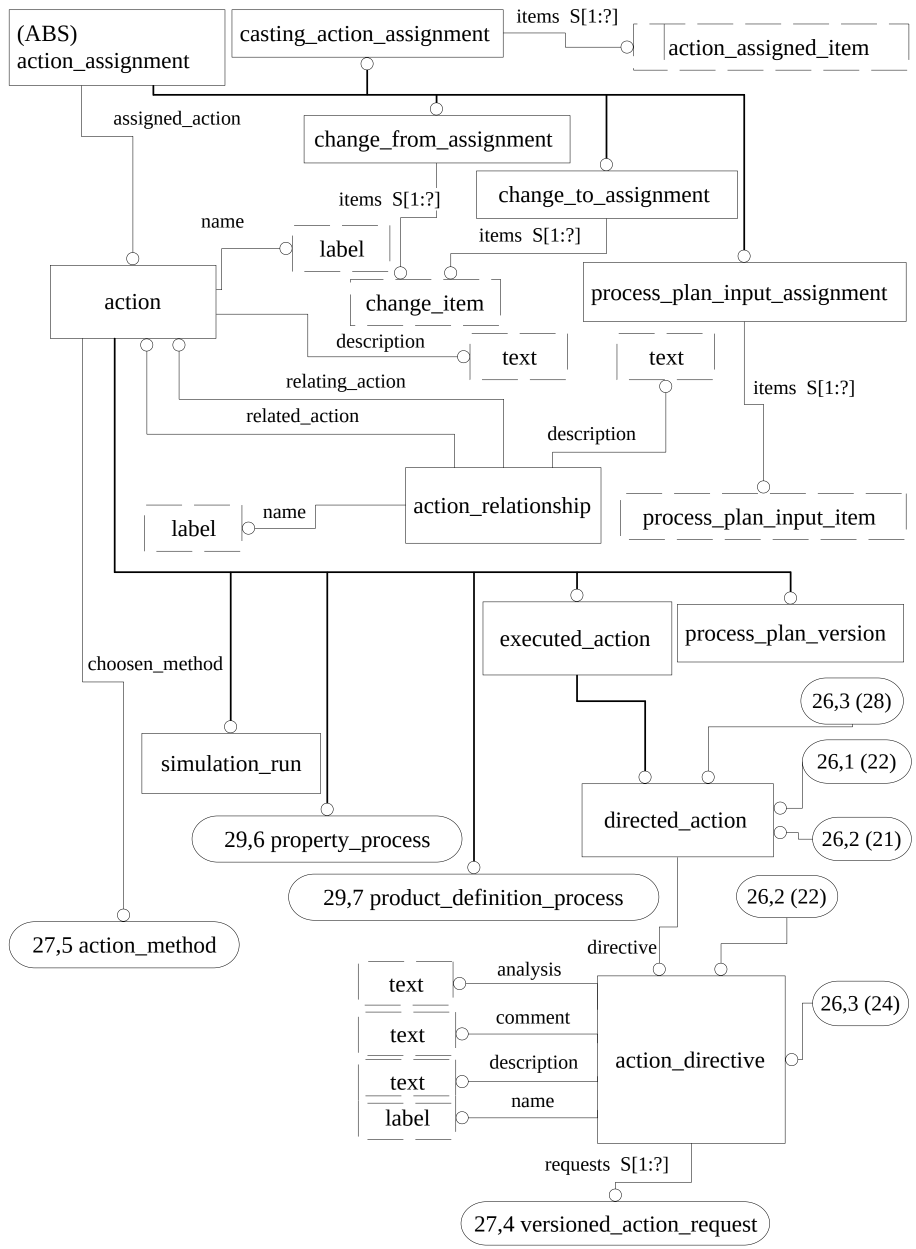


Figure H.26 – action - AIM EXPRESS-G diagram 26 of 34

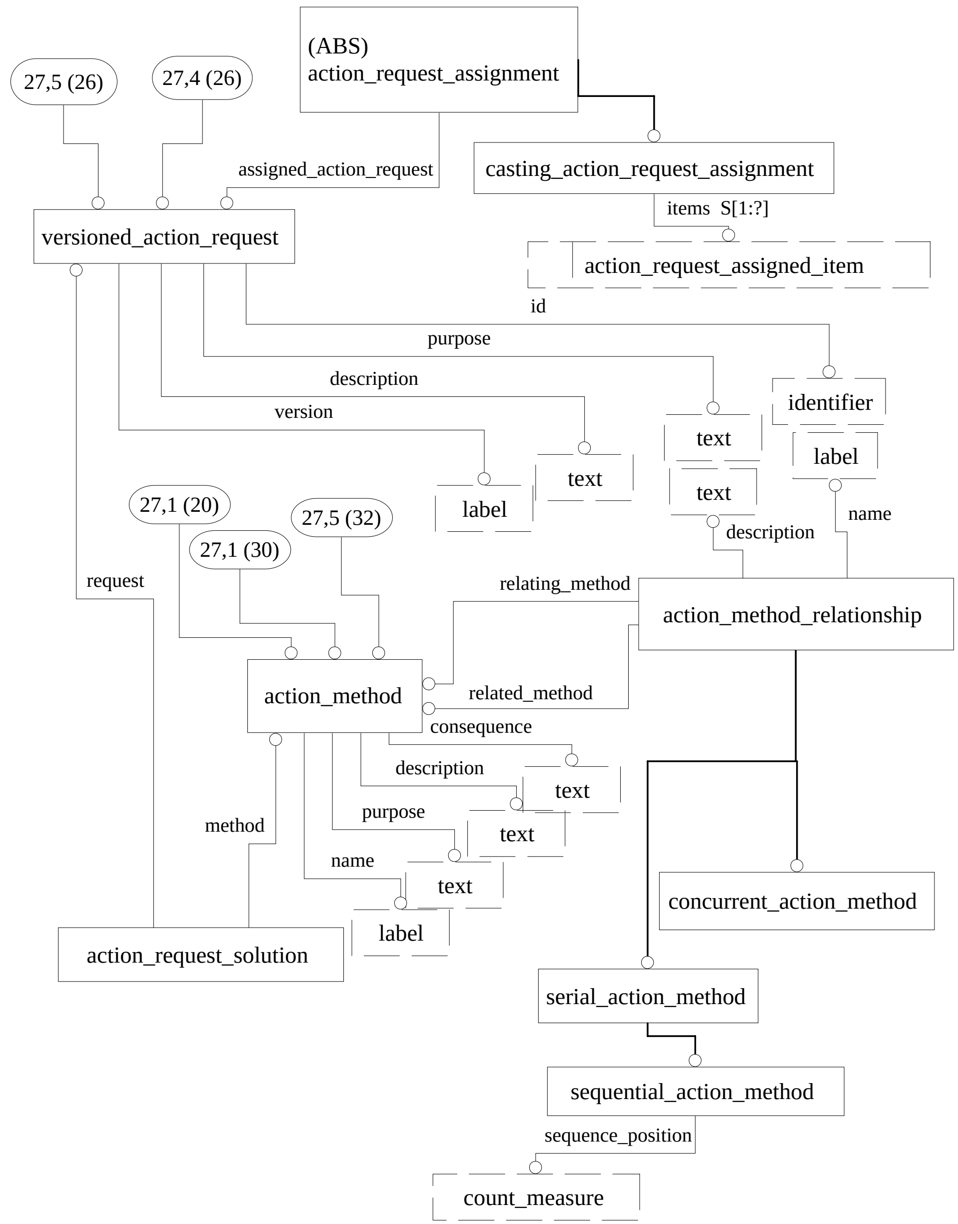


Figure H.27 – action request assignment - AIM EXPRESS-G diagram 27 of 34

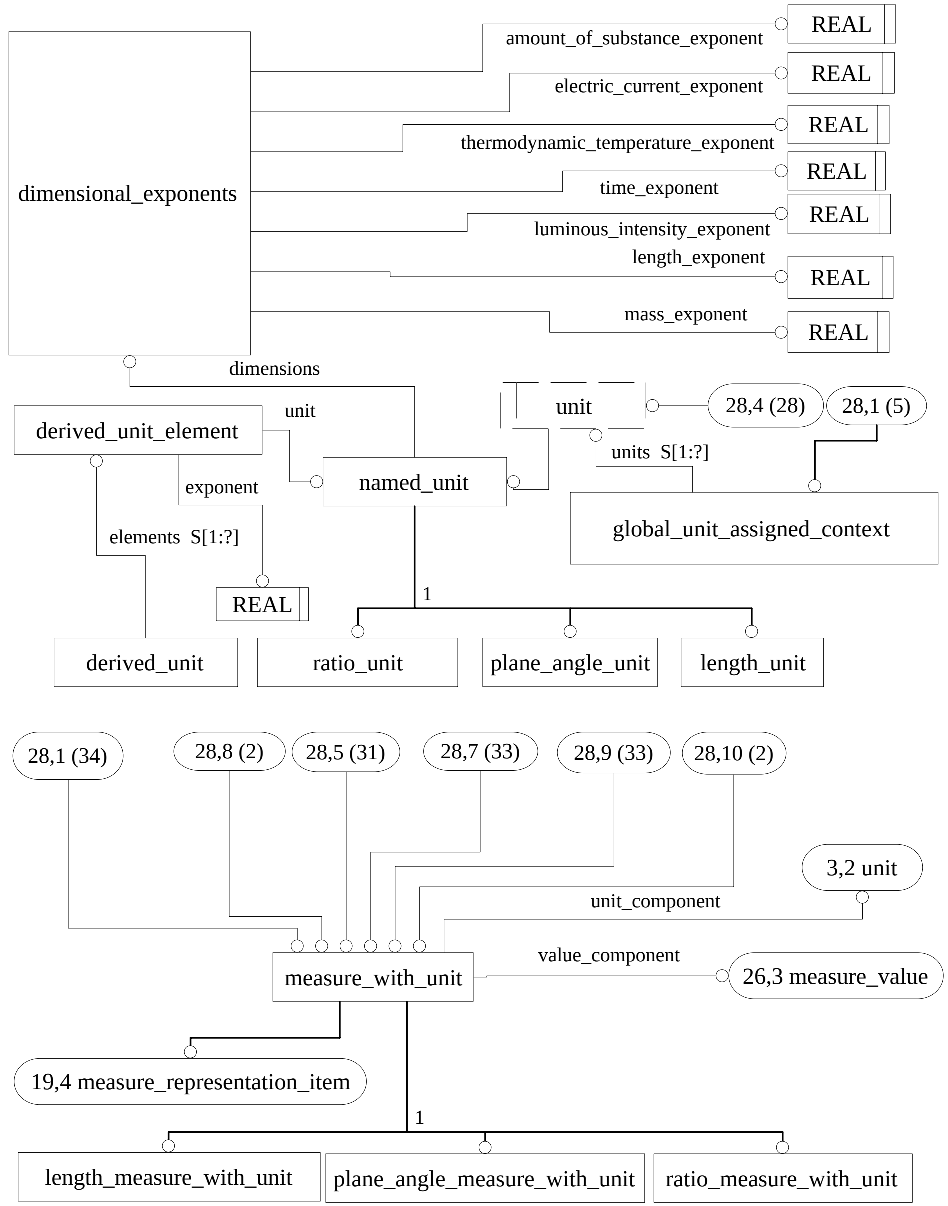


Figure H.28 – units - AIM EXPRESS-G diagram 28 of 34

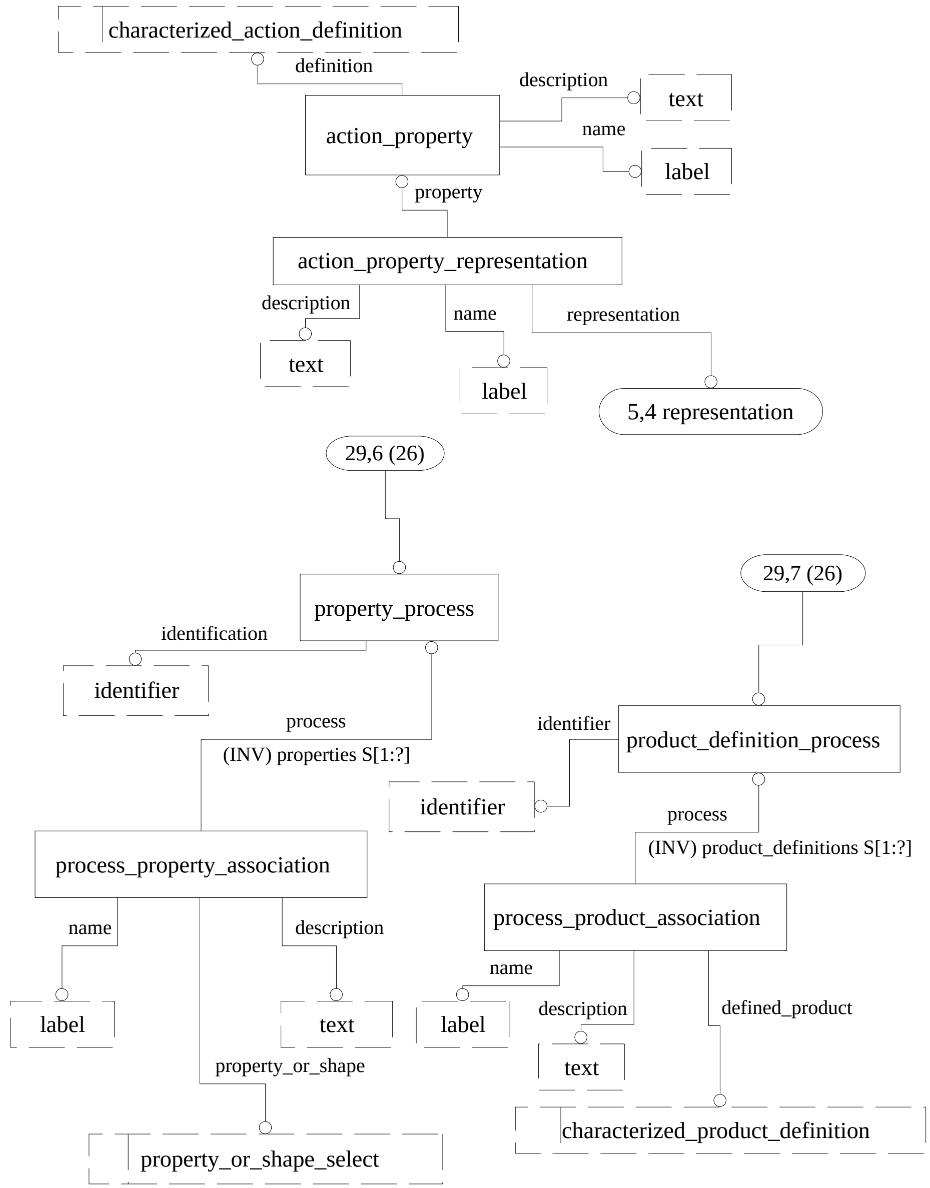


Figure H.29 – property process - AIM EXPRESS-G diagram 29 of 34

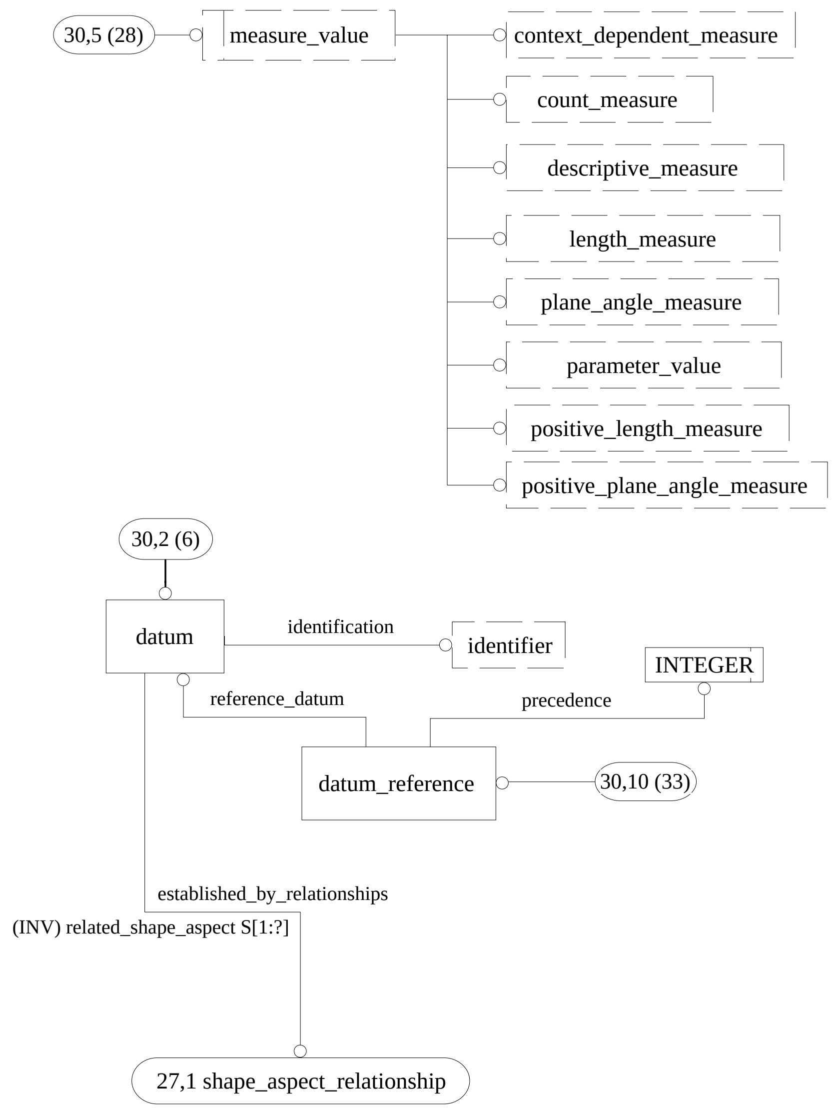


Figure H.30 – measures - AIM EXPRESS-G diagram 30 of 34

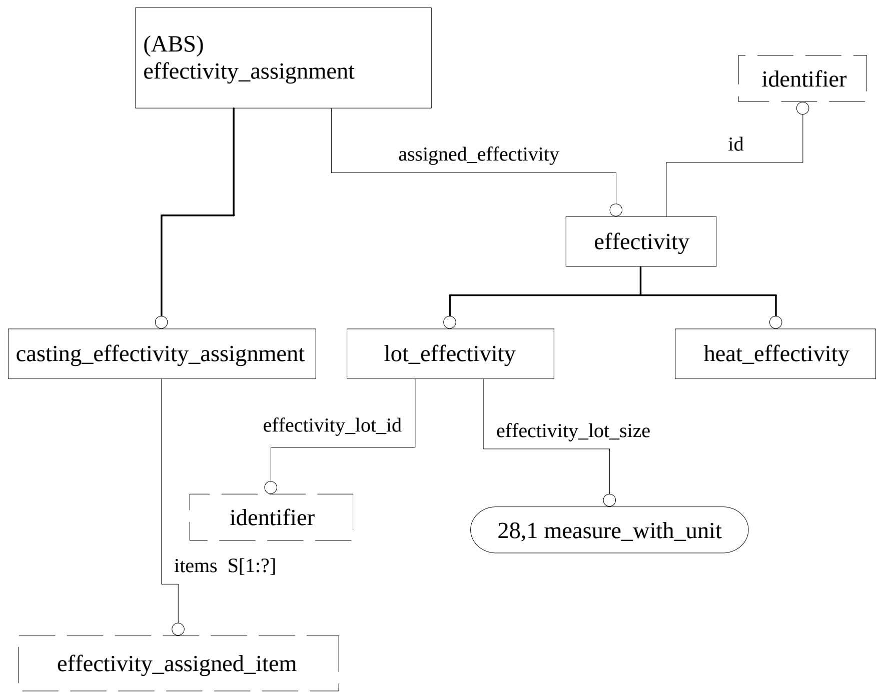


Figure H.31 – effectivity assignment - AIM EXPRESS-G diagram 31 of 34

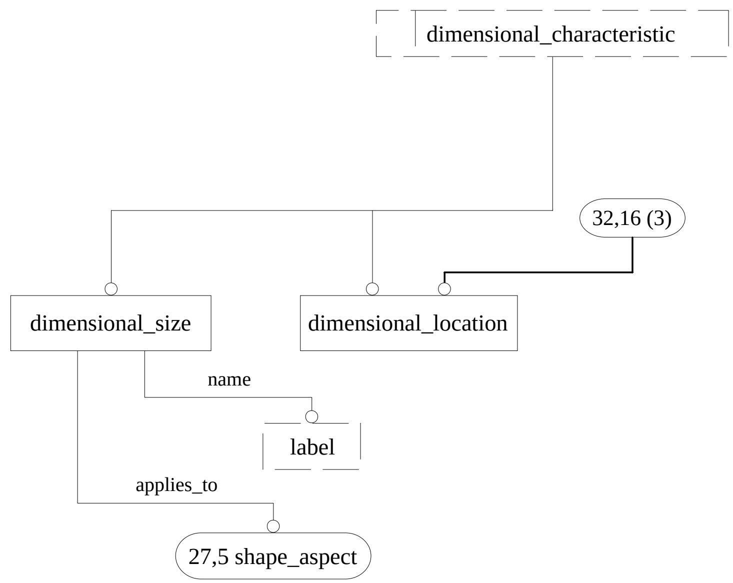


Figure H.32 – dimension - AIM EXPRESS-G diagram 32 of 34

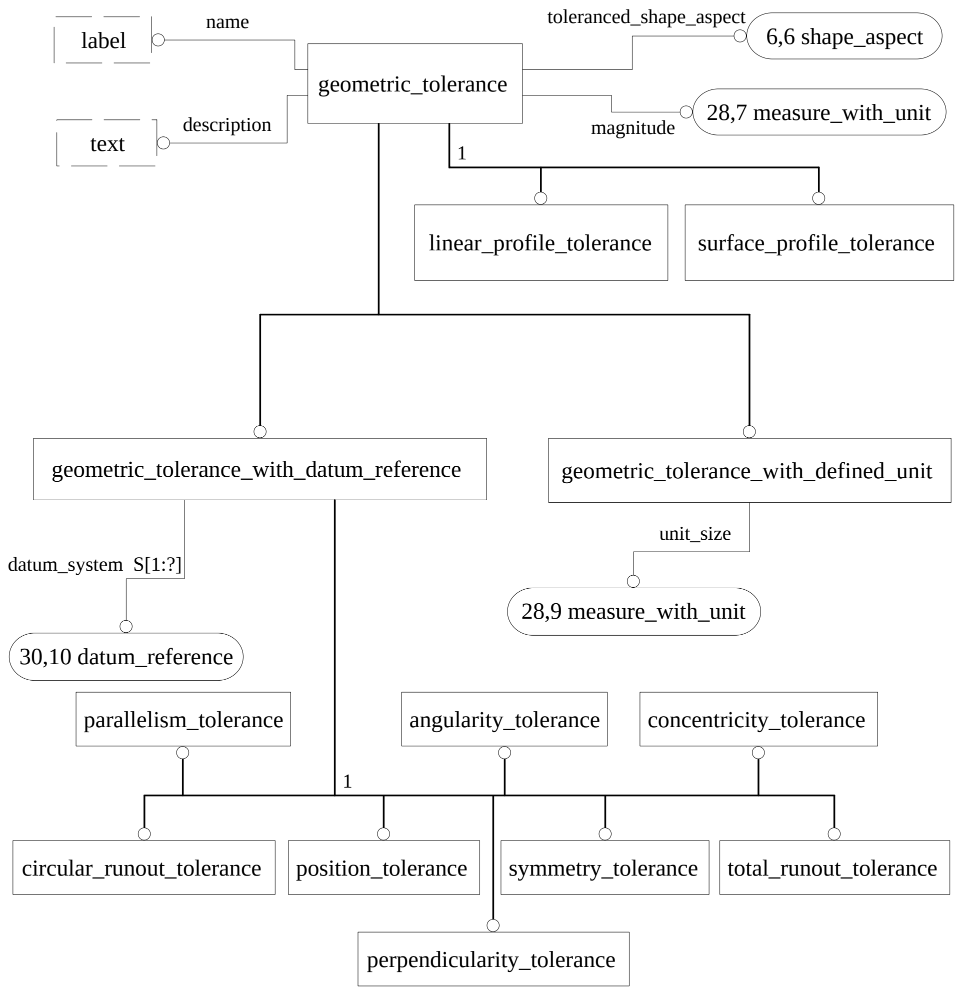


Figure H.33 – geometric tolerances - AIM EXPRESS-G diagram 33 of 34

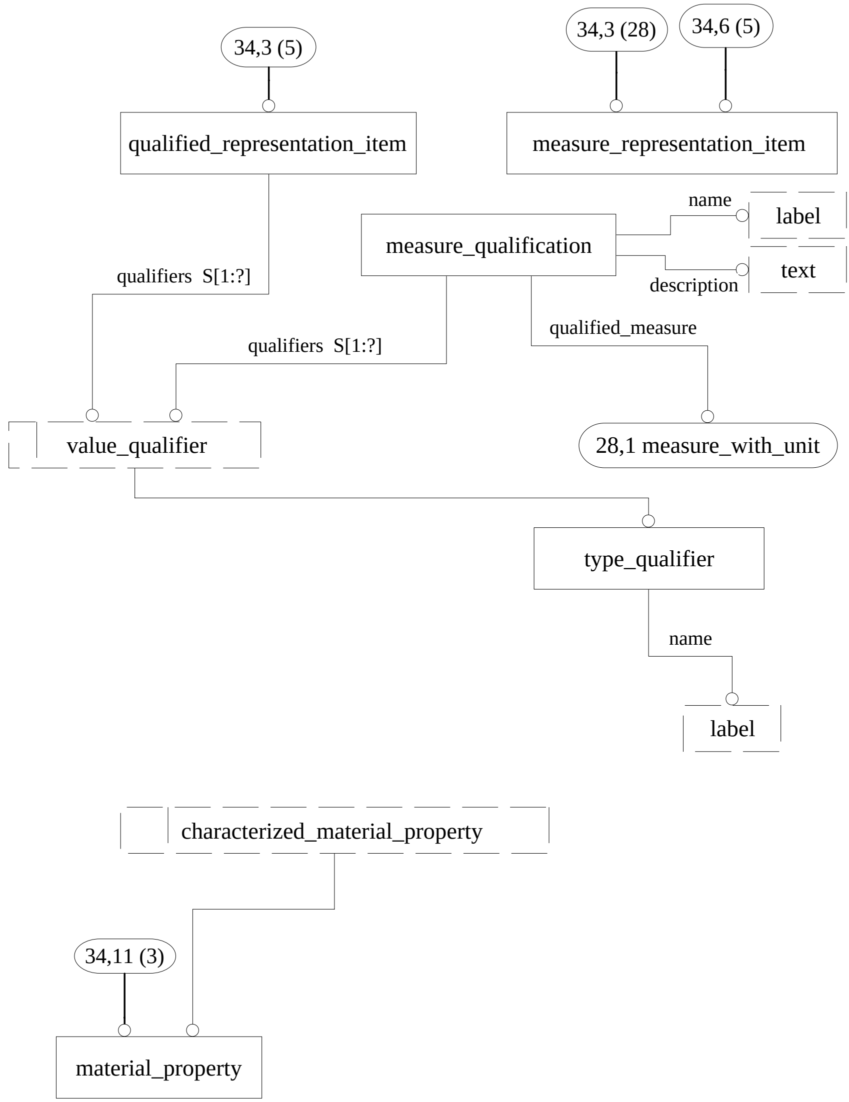


Figure H.34 – measure qualification - AIM EXPRESS-G diagram 34 of 34